

Foreword by **Nick Dallet** and **Jean Paoli**, Microsoft Corporation



Designing Forms for SharePoint and InfoPath

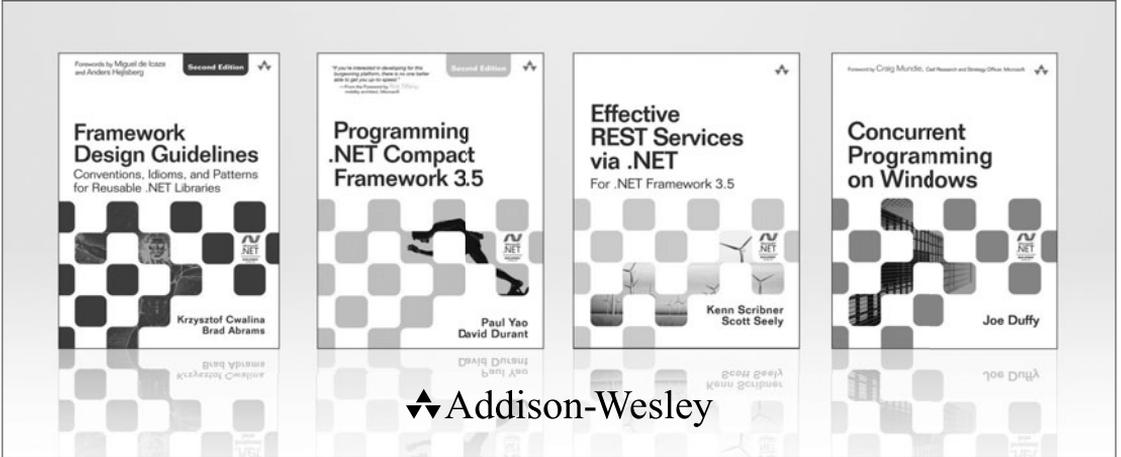
Using InfoPath Designer 2010



Scott Roberts
Hagen Green
Jessica Meats

Designing Forms for SharePoint and InfoPath

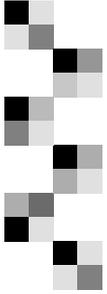
Microsoft® .NET Development Series



Visit informit.com/msdotnetseries for a complete list of available products.

The award-winning **Microsoft .NET Development Series** was established in 2002 to provide professional developers with the most comprehensive, practical coverage of the latest .NET technologies. Authors in this series include Microsoft architects, MVPs, and other experts and leaders in the field of Microsoft development technologies. Each book provides developers with the vital information and critical insight they need to write highly effective applications.





Designing Forms for SharePoint and InfoPath

Using InfoPath Designer 2010

- **Scott Roberts**
- **Hagen Green**
- **Jessica Meats**

◆ Addison-Wesley

Upper Saddle River, NJ • Boston • Indianapolis • San Francisco
New York • Toronto • Montreal • London • Munich • Paris • Madrid
Capetown • Sydney • Tokyo • Singapore • Mexico City

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The .NET_logo is either a registered trademark or trademark of Microsoft Corporation in the United States and/or other countries and is used under license from Microsoft.

Microsoft, Windows, Visual Basic, Visual C#, and Visual C++ are either registered trademarks or trademarks of Microsoft Corporation in the U.S.A. and/or other countries/regions.

The authors and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:

U.S. Corporate and Government Sales
(800) 382-3419
corpsales@pearsontechgroup.com

For sales outside the United States please contact:

International Sales
international@pearson.com

Visit us on the Web: informit.com/aw

Library of Congress Cataloging-in-Publication Data

Roberts, Scott, 1969-

Designing forms for SharePoint and InfoPath using InfoPath Designer 2010 / Scott Roberts,
Hagen Green, Jessica Meats.

p. cm.

Includes index.

ISBN 978-0-321-74360-2 (pbk. : alk. paper)

1. Microsoft InfoPath. 2. Microsoft SharePoint (Electronic resource) 3. Business—Forms—
Computer programs. I. Green, Hagen. II. Meats, Jessica. III. Title.

HF5371.R634 2011
651.7'4028553—dc22

2011008186

Copyright © 2011 Pearson Education, Inc.

All rights reserved. Printed in the United States of America. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, write to:

Pearson Education, Inc.
Rights and Contracts Department
501 Boylston Street, Suite 900
Boston, MA 02116
Fax: (617) 671-3447

ISBN-13: 978-0-321-74360-2

ISBN-10: 0-321-74360-1

Text printed in the United States on recycled paper at Edwards Brothers in Ann Arbor, Michigan.

First printing, May 2011

To my wife, Andrea, and my two sons, Sean and Bradley. It is due to your constant love and support that I was able to complete this project. Thank you for always being there for me no matter what project I undertake. I love you all a million times more than you could ever possibly know.

—Scott

To Jaime, my love and my life, for putting up with my two jobs and helping me stay strong through the end. And to my parents, Stuart and Christine, for their perseverance in love and for being the best parents anyone could ask for.

—Hagen

To Nick Borrett for a summer administration job that taught me the value of well-designed electronic forms. And to the friends who put up with the excuse, "Sorry, I've got a book to edit. Can we reschedule?"

—Jessica



This page intentionally left blank



Contents at a Glance

Figures xxi

Tables xlv

Foreword xlix

Preface liii

Acknowledgments lxi

About the Authors lxiii

Part I Designing Forms 1

- 1 Introduction to InfoPath 2010 3**
- 2 Basics of InfoPath Form Design 27**
- 3 Introducing Forms Services 79**
- 4 Working with Data 111**
- 5 Advanced Controls and Customization 185**
- 6 Adding Logic without Code 247**
- 7 Retrieving Data from External Sources 281**
- 8 Submitting Form Data 361**
- 9 Saving and Publishing 425**
- 10 Workflow 461**

**Part II Advanced Form Design 521****11 Security and Deployment 523****12 Creating Reports 619****13 Writing Code in InfoPath 679****14 Advanced Forms Services 791****15 Non-SharePoint Functionality 903****16 Importers and Exporters 963****Appendices 991****A Integration of InfoPath 2010 and SharePoint Server 2010
List Forms 993****B Further Reading 1021***Index 1027*



Contents

Figures xxi

Tables xlv

Foreword xlix

Preface liii

Acknowledgments lxi

About the Authors lxiii

PART I Designing Forms 1

1 Introduction to InfoPath 2010 3

What Is InfoPath? 3

InfoPath 2003 7

InfoPath 2003 Service Pack 1 12

InfoPath 2007 17

InfoPath 2010 22

What's Next? 25

2 Basics of InfoPath Form Design 27

Getting Started 27

What Is a Form Template Anyway? 28

Starting InfoPath Designer 29

Creating a New Blank Form Template 33

Designing the Layout of Your Form	35
<i>Themes</i>	39
<i>Using Controls</i>	40
<i>Input</i>	42
<i>Containers</i>	45
<i>Objects</i>	67
What's Next?	77
3 Introducing Forms Services	79
Getting Started	79
What Is InfoPath in the Browser?	80
<i>Why Use the Browser?</i>	82
<i>What Is Forms Services?</i>	83
Getting Familiar with Browser Forms	85
<i>Creating a New Form</i>	86
<i>Ribbon</i>	86
<i>Controls</i>	90
<i>Supported Web Browsers</i>	93
Designing a Browser-Enabled Form Template	95
<i>Design Once</i>	95
<i>Design Checker</i>	96
<i>Browser-Enabled Forms Without the Design Checker</i>	104
Getting Familiar with the Browser Form Experience	105
What's Next?	108
4 Working with Data	111
Getting Started	111
Data Binding	112
<i>Creating the Data Source Automatically</i>	112
<i>Understanding Data Source Details</i>	119
Data Source Field and Group Properties	121
<i>Name Property</i>	121
<i>Data Type Property</i>	124
<i>Cannot Be Blank Property</i>	127
<i>Repeating Property</i>	128

<i>Default Value Property</i>	132
<i>Data Source Details</i>	134
Manually Editing the Data Source	145
<i>Adding Nodes</i>	145
<i>Moving Nodes</i>	148
<i>Deleting Nodes</i>	151
<i>Referencing Nodes</i>	151
<i>Starting with Your Own Data Source</i>	153
Advanced Data Binding	167
<i>Understanding Data Binding</i>	168
<i>Design-Time Visuals</i>	178
Editing Default Template Data	180
What's Next?	183
5 Advanced Controls and Customization	185
Getting Started	185
Advanced Controls	186
<i>Hyperlink</i>	186
<i>Calculated Value</i>	189
<i>Vertical Label</i>	194
<i>Scrolling Region and Horizontal Region</i>	195
<i>Choice Group, Repeating Choice Group, and Choice Section</i>	197
<i>Repeating Recursive Section</i>	201
<i>Custom Controls</i>	202
Control Properties	204
<i>Control Formatting</i>	204
<i>Format Painter</i>	206
<i>Formatting Multiple Controls</i>	207
<i>Editing Control Properties</i>	209
<i>Data Tab</i>	210
<i>Display Tab</i>	216
<i>Size Tab</i>	222
<i>Advanced Tab</i>	227
<i>Creating Master/Detail Relationships</i>	232

- Advanced Customizations 238
 - Editing Default Values* 238
 - Customizing Commands* 242
- What's Next? 246

- 6 Adding Logic without Code 247**
 - Getting Started 247
 - Data Validation 247
 - Adding Data Validation to a Form Template* 248
 - Pattern Matching* 255
 - Conditional Formatting 258
 - Action Rules 264
 - Quick Rules* 270
 - Rule Inspector 272
 - Creating Multiple Views 276
 - What's Next? 278

- 7 Retrieving Data from External Sources 281**
 - Getting Started 281
 - Data Connections 282
 - Why Do We Need Them?* 282
 - Conceptually Understanding Data Connections and Data Sources* 283
 - All Data Connections Are Not Created Equal* 284
 - Creating a Data Connection* 285
 - SharePoint Libraries and Lists 286
 - XML Files 290
 - Databases 301
 - Setting Up a Connection to a Database* 302
 - Integrating the Database Connection into the Form Template* 312
 - SOAP Web Services 317
 - What Is a Web Service?* 317
 - Using a SOAP Web Service* 318
 - InfoPath Web Services* 320
 - Web Service Repository* 332

Extended Features of Web Services	333
<i>Searching with UDDI</i>	333
Data Connections in Browser Forms	334
<i>Data Connection Libraries</i>	335
<i>Security and Data Connection Libraries</i>	338
Secondary Data Source Binding	338
<i>Designing the Form Template</i>	339
<i>Filling Out the Form</i>	341
Offline Mode Query Support	343
List Box Controls Connected to Secondary Data Sources	348
<i>Showing Secondary Data in a List Box Control</i>	348
<i>Using the Main Data Source for List Box Items</i>	353
<i>Filtering List Box Items</i>	355
<i>Cascading Drop-Downs</i>	358
What's Next?	358
8 Submitting Form Data	361
Getting Started	361
Why Submit?	362
<i>Using the "Form-Only" Model to Disable Save</i>	363
<i>Issues with Data Validation and Submitting Forms</i>	364
Submitting Forms	367
<i>Introduction to Submit Data Connections</i>	368
<i>What Submit Method Is Best for My Form Template?</i>	369
<i>Submitting to a SharePoint List</i>	371
<i>Submitting to a SharePoint Library</i>	374
<i>Submitting via E-Mail</i>	377
<i>Submitting to a Database</i>	382
<i>Submitting to a Web Service</i>	384
<i>Including Text and Child Elements Only</i>	399
<i>XML Subtree, Including the Selected Element</i>	401
<i>Entire Form</i>	402
<i>Submitting Digitally Signed Form Data</i>	406
<i>Submitting to a Web Server via HTTP</i>	409
<i>Custom Submit Using Form Code</i>	412

Custom Submit Using Rules 413

Submitting to a Hosting Environment 419

Submit Options 420

What's Next? 423

9 Saving and Publishing 425

Getting Started 425

Saving and Publishing a Form Template with InfoPath Designer 426

Network Locations 429

SharePoint Server 432

Publishing a Form Template to Forms Services 442

E-Mail 446

Common Conveniences 449

Saving Templates as Exported Source Files 451

Previewing Form Templates 454

With Sample Data 455

With User Roles 456

Domain Simulation 458

Saving Form Data 458

What's Next? 459

10 Workflow 461

Getting Started 461

Workflow with Microsoft Office SharePoint Server 2010 462

Using InfoPath Forms with SharePoint Designer 470

User Roles 479

Creating User Roles 481

Determining a User's Role When Filling Out a Form 485

User Roles in Action 486

Role-Based Views 497

Workflow with InfoPath E-Mail Forms 499

Designing and Using InfoPath E-Mail Forms 501

Creating Rules for InfoPath E-Mail Forms 505

Storing Received Forms in Outlook Folders 507

Filling Out an InfoPath E-Mail Form 509



<i>Sorting, Grouping, and Filtering Responses</i>	513
<i>Merging and Exporting InfoPath E-Mail Forms</i>	515
<i>Customizing E-Mail Support for a Form Template</i>	517
What's Next?	519

PART II Advanced Form Design 521

11 Security and Deployment 523

Getting Started	523
Introduction to InfoPath Security	525
InfoPath Security Levels	526
<i>Designing Security into Form Templates</i>	527
<i>Restricted Security Level</i>	529
<i>Domain Security Level</i>	535
<i>Automatic Security Level</i>	550
<i>Full Trust Security Level</i>	552
Trust and Designer Lockdown	566
<i>Trust Center</i>	566
<i>Designer Lockdown</i>	572
Digital Signatures	574
<i>Using Digitally Signed Data in Forms</i>	576
<i>Allowing Only One Signature</i>	594
<i>Using Independent Signatures (Co-signing)</i>	598
<i>Setting Up Counter-Signatures</i>	599
<i>Digital Signatures in the XML Data and Schema</i>	601
<i>Signature Line Control</i>	605
Information Rights Management	607
<i>Permission on Forms</i>	608
<i>Permission with Document Libraries</i>	614
What's Next?	618

12 Creating Reports 619

Getting Started	619
Merging Forms	620
<i>Merging Forms in InfoPath</i>	621
<i>Design Recommendations for Merging Forms</i>	625

<i>Customizing Merge Behavior in InfoPath Designer</i>	631
<i>Custom Merge XSL (Advanced)</i>	645
Printing	652
<i>Print Views</i>	652
<i>Headers and Footers</i>	655
<i>Multiple View Printing</i>	657
<i>Word Print Views</i>	662
<i>Print Views in Browser-Enabled Forms</i>	668
Exporting Forms	670
What's Next?	677
13 Writing Code in InfoPath	679
Getting Started	679
Writing Code Behind a Form	680
<i>Settings Related to Adding Code</i>	681
<i>Adding Code to a Form Template</i>	686
<i>Filling Out and Debugging a Form with Code</i>	688
The InfoPath Object Model	691
<i>Form Events</i>	692
<i>XML Data Events</i>	693
<i>Using XPathNavigator Objects</i>	727
<i>Registering Event Handlers</i>	731
<i>Script in the Custom Task Pane</i>	733
Programming InfoPath . . . in Action!	739
<i>The MOI Consulting Request Form</i>	739
<i>Filling Out the MOI Consulting Request Form</i>	739
<i>Designing the MOI Consulting Request Form</i>	746
Form Code	781
<i>Executing Form Code in the Browser</i>	782
<i>Circumventing Browser-Enabled Limitations</i>	783
<i>Detecting the Browser or the InfoPath Client</i>	785
<i>Form Code Compatibility with Forms Services</i>	787
What's Next?	790



14	Advanced Forms Services	791
	Getting Started	791
	Controls and Browser Optimizations	792
	<i>Postback Settings</i>	796
	<i>The Update Button</i>	797
	<i>Designing Accessible Forms</i>	800
	Advanced Publishing and Administration	801
	<i>Advanced Publishing</i>	802
	<i>Managing Form Templates</i>	819
	<i>Configuring InfoPath Forms Services</i>	821
	<i>Configuring Services</i>	822
	Data Connections	842
	<i>Data Connections Administration Settings</i>	844
	<i>Centrally Managed Connection Library</i>	849
	<i>Authentication Considerations</i>	857
	<i>E-Mail Data Connections</i>	862
	Hosting the InfoPath Form Control in a Web Browser	864
	<i>Host to InfoPath Communication</i>	873
	<i>InfoPath to Host Communication</i>	881
	Performance Tips and Best Design Practices	888
	<i>Form Template Deployment</i>	889
	<i>Views</i>	889
	<i>Form Code</i>	890
	<i>Reduction of Form Postbacks</i>	891
	<i>Data-Heavy Features</i>	892
	<i>Data Connections</i>	893
	<i>Form View State</i>	894
	<i>Miscellaneous Performance Tips</i>	895
	<i>Performance Monitoring</i>	896
	<i>Health Monitoring</i>	900
	What's Next?	901
15	Non-SharePoint Functionality	903
	Getting Started	903

Hosting Scenarios	904
Document Information Panel	905
Creating an InfoPath Host Application	916
<i>Creating an InfoPath Host Application in .NET</i>	917
<i>Host to InfoPath Communication</i>	929
<i>Handling Events from the Form</i>	939
<i>Using the Host Property</i>	943
<i>Submitting a Form to the Host</i>	946
Installing and Using ActiveX Controls	950
<i>Adding an ActiveX Control to the Controls Task Pane</i>	950
<i>Property Pages</i>	959
Building Custom Controls for InfoPath Using ActiveX	960
What's Next?	962
16 Importers and Exporters	963
Getting Started	963
Built-in Form Importers	964
<i>Importing a Form into InfoPath</i>	967
<i>Fixing the Imported Form</i>	972
<i>Post-Import Warnings</i>	974
Creating Your Own Form Importers and Exporters	976
<i>InfoPath Import/Export Framework</i>	976
<i>How Post-Import Warnings Work</i>	988
What's Next?	990
Appendices	991
A Integration of InfoPath 2010 and SharePoint Server 2010	
List Forms	993
Building or Updating a Site	993
Capabilities of InfoPath and SharePoint List Forms	994
<i>New SharePoint List Form Files</i>	995
<i>Lists to Be Created for Mashup Page</i>	996
<i>Creating SharePoint Lists</i>	997
<i>Publishing Forms</i>	997
<i>Creating Example SharePoint Lists</i>	997

Editing SharePoint Lists Forms	998
<i>Editing SharePoint List</i>	998
<i>Adding a New Column</i>	999
<i>Adding a Field to SharePoint List Form</i>	1002
<i>Removing a Field from SharePoint List Form</i>	1003
<i>Adding List Data to SharePoint Lists</i>	1005
Building a Mashup Page	1006
<i>SharePoint Web Parts</i>	1006
<i>Creating Status List and Indicators</i>	1007
<i>Creating a New Library and Page for the Mashup Page</i>	1010
<i>Adding and Configuring Web Parts on a Wiki Page</i>	1011
Exporting Data to Excel or Access for Simple Reporting	1018
Summary	1019
B Further Reading	1021
Chapter 2: Basics of InfoPath Form Design	1021
Chapter 4: Working with Data	1021
Chapter 5: Advanced Controls and Customization	1022
Chapter 7: Retrieving Data from External Sources	1022
Chapter 8: Submitting Form Data	1022
Chapter 9: Saving and Publishing	1023
Chapter 10: Workflow	1023
Chapter 11: Security and Deployment	1023
Chapter 12: Creating Reports	1024
Chapter 13: Writing Code in InfoPath	1024
Chapter 14: Advanced Forms Services	1025
Chapter 15: Non-SharePoint Functionality	1025
General Reference Material	1026
<i>Index</i>	1027

This page intentionally left blank



Figures

- FIGURE P.1:** *Open with Form Template Dialog* lvii
- FIGURE P.2:** *Dialog shown when InfoPath cannot find the Visual C# project with the form code* lix
-
- FIGURE 1.1:** *Controls task pane in InfoPath 2003* 8
- FIGURE 1.2:** *Data Source task pane in InfoPath 2003* 9
- FIGURE 1.3:** *Insert Layout Table toolbar item* 13
- FIGURE 1.4:** *Tables toolbar* 13
- FIGURE 1.5:** *Fill Out a Form dialog in InfoPath 2003 SP1* 17
- FIGURE 1.6:** *Filling out a browser-enabled form template in Internet Explorer* 19
- FIGURE 1.7:** *The table layout tab* 23
- FIGURE 1.8:** *The Rules task pane prior to rules being created* 24
-
- FIGURE 2.1:** *Opening screen* 30
- FIGURE 2.2:** *New blank form* 35
- FIGURE 2.3:** *Table Tools Layout tab* 36
- FIGURE 2.4:** *Employment application form template with title added* 37
- FIGURE 2.5:** *Employment application form template with cells for first and last names* 38
- FIGURE 2.6:** *Employment application form template with a row for address information* 39
- FIGURE 2.7:** *Page Design tab* 40

- FIGURE 2.8:** *Controls task pane* 41
- FIGURE 2.9:** *Employment application form template after adding standard controls* 46
- FIGURE 2.10:** *Section control selected* 47
- FIGURE 2.11:** *Preview of the employment application form* 48
- FIGURE 2.12:** *Employment application form template with Optional Section controls* 50
- FIGURE 2.13:** *Optional controls when previewing the form* 51
- FIGURE 2.14:** *Optional controls after clicking on the link, Click here to insert* 51
- FIGURE 2.15:** *Context menu for the Optional Section control when filling out a form* 52
- FIGURE 2.16:** *Employment application form template with Repeating Section control* 53
- FIGURE 2.17:** *Repeating Section control when previewing the form* 54
- FIGURE 2.18:** *Context menu for Repeating Section control when previewing the form* 54
- FIGURE 2.19:** *Three instances of the Repeating Section control inserted into the form* 55
- FIGURE 2.20:** *Insert Repeating Table dialog* 57
- FIGURE 2.21:** *Repeating Table control in InfoPath Designer* 57
- FIGURE 2.22:** *Change To menu* 58
- FIGURE 2.23:** *Repeating Table control after changing from a Repeating Section control* 58
- FIGURE 2.24:** *Repeating Table control when previewing the form* 59
- FIGURE 2.25:** *Horizontal Repeating Table control in Designer* 61
- FIGURE 2.26:** *Horizontal Repeating Table control when filling out a form* 61
- FIGURE 2.27:** *Insert Master/Detail dialog* 62
- FIGURE 2.28:** *Master and detail controls in Designer* 63
- FIGURE 2.29:** *Master/Detail control in the application review form when filling out a form* 63
- FIGURE 2.30:** *Master/Detail control showing more applicant data* 65
- FIGURE 2.31:** *Binding design-time visual showing the name of the control* 65
- FIGURE 2.32:** *List controls in Designer* 66
- FIGURE 2.33:** *List controls when filling out a form* 66
- FIGURE 2.34:** *Multiple-Selection List Box control when filling out a form* 67



- FIGURE 2.35:** *File Attachment control in Designer* 68
- FIGURE 2.36:** *File Attachment control when filling out a form* 68
- FIGURE 2.37:** *File Attachment control with résumé file attached* 68
- FIGURE 2.38:** *File Attachment control commands* 69
- FIGURE 2.39:** *Insert Picture Control dialog* 70
- FIGURE 2.40:** *Picture control when filling out the form* 70
- FIGURE 2.41:** *Picture control with picture inserted* 71
- FIGURE 2.42:** *Ink Picture control that contains text entered with a Tablet PC stylus* 72
- FIGURE 2.43:** *Button control* 73
- FIGURE 2.44:** *The Picture Button control when first inserted* 73
- FIGURE 2.45:** *The Control Tools Properties tab* 73
- FIGURE 2.46:** *Insert Calculated Value dialog* 74
- FIGURE 2.47:** *Insert Vertical Label dialog* 74
- FIGURE 2.48:** *Vertical Label control in Designer* 75
- FIGURE 2.49:** *Hyperlink control* 75
- FIGURE 2.50:** *Insert Hyperlink dialog* 76
- FIGURE 2.51:** *Signature Line control in InfoPath Filler* 76
- FIGURE 2.52:** *The Sign dialog* 77
-
- FIGURE 3.1:** *Filling out the status report form in InfoPath* 81
- FIGURE 3.2:** *Filling out the status report form in the browser by using Forms Services* 81
- FIGURE 3.3:** *Message indicating that a feature cannot be used in a browser-enabled form template* 85
- FIGURE 3.4:** *Message indicating that a feature can be used in a browser-enabled form template but is not active in the browser* 85
- FIGURE 3.5:** *Toolbar shown when filling out a form in the browser* 86
- FIGURE 3.6:** *Configuring toolbars for a browser-enabled form template filled out in the browser* 88
- FIGURE 3.7:** *Saving a form in the browser* 89
- FIGURE 3.8:** *Editing a Rich Text Box control in the browser* 93
- FIGURE 3.9:** *Commonality of InfoPath and Forms Services feature sets* 96
- FIGURE 3.10:** *MOI Consulting feedback form* 97
- FIGURE 3.11:** *Compatibility settings in the Form Options dialog* 98

- FIGURE 3.12:** *Errors and messages in the Design Checker after making the feedback form template browser-compatible* 99
- FIGURE 3.13:** *Browser-compatibility message within the Digital Signatures category under the Form Options dialog* 100
- FIGURE 3.14:** *Remake of the Overall Satisfaction table without vertical text* 102
- FIGURE 3.15:** *Filling out the MOI Consulting feedback form in the browser* 106
-
- FIGURE 4.1:** *Clicking on a control in the view to select the bound control in the data source* 113
- FIGURE 4.2:** *A Person/Group Picker control bound to group1 in the data source* 114
- FIGURE 4.3:** *Warning dialog when changing the name of a data source field or group for a published form* 123
- FIGURE 4.4:** *Data type validation error that occurs when a field contains invalid data* 126
- FIGURE 4.5:** *Warning when saving a form with validation errors* 127
- FIGURE 4.6:** *Validation error for the Cannot be blank property, delineated by a red asterisk* 128
- FIGURE 4.7:** *Repeating field4, which is bound to the Numbered List control and repeats in group1* 130
- FIGURE 4.8:** *Airline form with default value for FlightType as "Roundtrip"* 132
- FIGURE 4.9:** *The default value calculation button* 133
- FIGURE 4.10:** *The Insert Formula dialog, with a formula that sets this field's default value to today's date plus 45 days* 133
- FIGURE 4.11:** *Data source details for the FlightType field bound to the option buttons in the airline form* 134
- FIGURE 4.12:** *Label for the node PurposeOfVisit, created when inserting the control into the view* 148
- FIGURE 4.13:** *Move Field or Group dialog used to move an existing field or group* 149
- FIGURE 4.14:** *Validation error that appears when opening a form created before the FlightType-FlightDate node swap* 150
- FIGURE 4.15:** *Setting the Employee group to repeat and reference itself* 152
- FIGURE 4.16:** *Filling out the MOI Company's employee information form* 153

- FIGURE 4.17:** *Design a form from an existing XML document or schema instead of creating the data source from scratch* 154
- FIGURE 4.18:** *Selecting the location of an XML document or schema* 155
- FIGURE 4.19:** *Choosing whether to include data from a selected XML file as the form template default data* 156
- FIGURE 4.20:** *The first prompt when starting from an ambiguous schema* 159
- FIGURE 4.21:** *Disambiguating the data source by using the Edit Settings dialog* 160
- FIGURE 4.22:** *Prompt that appears when starting from a schema with multiple top-level elements* 160
- FIGURE 4.23:** *The Type drop-down menu, with an option for adding a complete XML Schema or XML document* 164
- FIGURE 4.24:** *Data source created from the XML document* 166
- FIGURE 4.25:** *Text Box Binding dialog that appears when Automatically create data source is disabled or Change Binding is used from a control* 169
- FIGURE 4.26:** *Changing binding via the context menu by right-clicking on any control that can be bound* 169
- FIGURE 4.27:** *Context menu of the most popular control bindings* 172
- FIGURE 4.28:** *Dialog that results from clicking More on the menu shown in Figure 4.27* 172
- FIGURE 4.29:** *Inserting field1 in the data source. The context of a drop affects available control binding options.* 173
- FIGURE 4.30:** *An absurd control nesting: Attribute contains DocumentElement, which contains Attribute* 175
- FIGURE 4.31:** *NumberOfChildren attribute field, which is bound to both the Optional Section and the Drop-Down List Box inside of the Optional Section* 176
- FIGURE 4.32:** *NumberOfChildren attribute is nonexistent in the form and XML data* 176
- FIGURE 4.33:** *NumberOfChildren attribute field inserted in the form and XML with a default value of 1* 176
- FIGURE 4.34:** *A simple repeating field: field1* 177
- FIGURE 4.35:** *Three controls multiply bound to field1* 177
- FIGURE 4.36:** *Critical design-time visual on an unbound Text Box control* 180
- FIGURE 4.37:** *Cutaway of the Edit Default Values dialog when Choice exists in the data source* 181

- FIGURE 4.38:** *Cutaway of the Edit Default Values dialog when Choice exists in the data source* 182
- FIGURE 5.1:** *Insert Hyperlink dialog* 187
- FIGURE 5.2:** *Select a Field or Group dialog* 188
- FIGURE 5.3:** *Hyperlink control in InfoPath Designer* 188
- FIGURE 5.4:** *Insert Hyperlink dialog in InfoPath Filler* 189
- FIGURE 5.5:** *Insert Calculated Value dialog* 190
- FIGURE 5.6:** *Insert Formula dialog after choosing a field* 190
- FIGURE 5.7:** *Insert Formula dialog after checking the Edit XPath checkbox* 191
- FIGURE 5.8:** *Insert Function dialog* 191
- FIGURE 5.9:** *Repeating Table control for expenses* 192
- FIGURE 5.10:** *Data source for the expense report example* 192
- FIGURE 5.11:** *Insert Formula dialog after inserting the sum function* 192
- FIGURE 5.12:** *Expense report form during preview* 193
- FIGURE 5.13:** *Insert Vertical Label dialog* 194
- FIGURE 5.14:** *Vertical Label control* 194
- FIGURE 5.15:** *Scrolling Region control in InfoPath Designer* 195
- FIGURE 5.16:** *Application review form with Scrolling Region when filling out the form* 196
- FIGURE 5.17:** *Horizontal Region controls in InfoPath Designer* 197
- FIGURE 5.18:** *Choice Group control in InfoPath Designer* 198
- FIGURE 5.19:** *Employment application form with Choice Group control* 200
- FIGURE 5.20:** *Default Choice Section with context menu commands* 200
- FIGURE 5.21:** *Employment application form after replacing the default Choice Section* 200
- FIGURE 5.22:** *Repeating Recursive Section in InfoPath Designer* 202
- FIGURE 5.23:** *Repeating Recursive Section when filling out the employee information form* 203
- FIGURE 5.24:** *Selected Text Box control* 205
- FIGURE 5.25:** *Format Painter in the Home tab* 206
- FIGURE 5.26:** *Format Painter cursor* 206
- FIGURE 5.27:** *Text Settings tab in the View Properties dialog* 208
- FIGURE 5.28:** *Properties dialog for the Text Box control* 209



- FIGURE 5.29:** *Dialog that warns about data loss when changing field names* 211
- FIGURE 5.30:** *Binding design-time visual for the FirstName Text Box control* 212
- FIGURE 5.31:** *Integer Format dialog for Whole Number (integer) data types* 213
- FIGURE 5.32:** *Field for minimum salary desired formatted as currency* 213
- FIGURE 5.33:** *Properties dialog for the Optional Section for the minimum salary desired* 215
- FIGURE 5.34:** *Display tab of the Text Box Properties dialog* 217
- FIGURE 5.35:** *Placeholder text in a Text Box control* 218
- FIGURE 5.36:** *Display tab for the Rich Text Box control* 220
- FIGURE 5.37:** *Insert Picture dialog* 221
- FIGURE 5.38:** *Size tab of the Text Box Properties dialog* 223
- FIGURE 5.39:** *Text Box control with inner text not aligned to outer text* 224
- FIGURE 5.40:** *Text Box control with inner text aligned to outer text* 225
- FIGURE 5.41:** *Properties dialog for resizing multiple controls* 226
- FIGURE 5.42:** *Properties tab with multiple controls selected* 226
- FIGURE 5.43:** *Advanced tab for the Hyperlink control* 227
- FIGURE 5.44:** *Advanced tab of the Text Box Properties dialog showing the Input recognition section* 228
- FIGURE 5.45:** *Input Scope dialog showing standard input scopes* 229
- FIGURE 5.46:** *Input Scope dialog showing custom input scopes* 230
- FIGURE 5.47:** *New Input Scope dialog showing Phrase List option* 231
- FIGURE 5.48:** *Sample data source for the Master/Detail example* 233
- FIGURE 5.49:** *Master/Detail tab for a Repeating Table set as the master control* 235
- FIGURE 5.50:** *Master/Detail tab for a Repeating Section set as a detail control* 235
- FIGURE 5.51:** *Master/Detail control when filling out a form* 237
- FIGURE 5.52:** *Section Properties dialog for an Optional Section control* 239
- FIGURE 5.53:** *Edit Default Values dialog for an Optional Section control* 240
- FIGURE 5.54:** *Repeating Section Properties dialog* 241
- FIGURE 5.55:** *Section Properties dialog for a Repeating Section fragment* 242
- FIGURE 5.56:** *Repeating Section Properties dialog showing three sections—Apples, Oranges, and Bananas* 243
- FIGURE 5.57:** *Filling out the grocery shopping list* 243
- FIGURE 5.58:** *Section Commands dialog used to customize commands* 245
- FIGURE 5.59:** *Default commands for a Choice Section when filling out a form* 246

- FIGURE 6.1:** *Rules pane* 249
- FIGURE 6.2:** *Rules pane with a validation rule added* 250
- FIGURE 6.3:** *Condition dialog* 251
- FIGURE 6.4:** *Condition dialog with multiple conditions* 253
- FIGURE 6.5:** *Rules pane with validation rule defined* 254
- FIGURE 6.6:** *Inline alert for a data validation error* 254
- FIGURE 6.7:** *Full error description for a data validation error* 255
- FIGURE 6.8:** *Data Validation dialog listing two conditions to validate the total number of guests* 256
- FIGURE 6.9:** *Data Entry Pattern dialog* 257
- FIGURE 6.10:** *Red asterisk error* 257
- FIGURE 6.11:** *Custom pattern* 258
- FIGURE 6.12:** *Rules pane with formatting rule added* 260
- FIGURE 6.13:** *Rules pane with condition, where StockPrice is less than or equal to \$18.00* 261
- FIGURE 6.14:** *Rules pane specifying the four conditions needed to set the background color to yellow* 262
- FIGURE 6.15:** *Rules pane showing all three rules* 263
- FIGURE 6.16:** *Defining a rule so that four actions are always run when the form is submitted* 265
- FIGURE 6.17:** *Choosing to add a new action rule* 265
- FIGURE 6.18:** *Specifying the action for a rule to run when a user clicks the Button control* 266
- FIGURE 6.19:** *Dialog displayed when an action on a rule fails to execute when the user fills out the form* 269
- FIGURE 6.20:** *The Add Rules menu when a text box is selected* 270
- FIGURE 6.21:** *The Rule Detail dialog for an Is less than rule* 271
- FIGURE 6.22:** *Form template for the Rule Inspector sample* 273
- FIGURE 6.23:** *Rule Inspector showing the logic for the entire form template* 274
- FIGURE 6.24:** *Detailed view of the DataValidation node* 275
- FIGURE 6.25:** *Views controls in Page Design tab* 276
- FIGURE 6.26:** *Add View dialog* 277
- FIGURE 6.27:** *View Properties dialog* 277

- FIGURE 7.1:** *Relationships between a form and its external data sources* 283
- FIGURE 7.2:** *Available Form Templates, the starting point for creating a main query connection* 286
- FIGURE 7.3:** *Creating a secondary query connection for a form template* 287
- FIGURE 7.4:** *Pictures library on SharePoint Foundation* 288
- FIGURE 7.5:** *The libraries and lists available for querying from the SharePoint server* 289
- FIGURE 7.6:** *Choosing which fields from the SharePoint list you want to use for the data connection* 290
- FIGURE 7.7:** *Filling out the music and photos form. The list of photos comes from the Pictures library on SharePoint.* 291
- FIGURE 7.8:** *Data to fill into a form and save to Sample.xml, to be read later by using an XML file adapter* 293
- FIGURE 7.9:** *Entering the file location to use when creating a secondary connection to XML data* 294
- FIGURE 7.10:** *XML file connection settings to opt in or out of including the data as a file in your template* 295
- FIGURE 7.11:** *Final page of the Data Connection Wizard for an XML file* 296
- FIGURE 7.12:** *Data Connections dialog after finishing the XML file Data Connection Wizard* 297
- FIGURE 7.13:** *Fields task pane showing secondary data source* 298
- FIGURE 7.14:** *Clicking on the Button to query the Sample data connection* 300
- FIGURE 7.15:** *Data Connection Wizard when creating a new database connection* 304
- FIGURE 7.16:** *Connecting to SQL Server after clicking Select Database* 305
- FIGURE 7.17:** *Select Database and Table dialog. Choose the Northwind database and the Customers table. The top two items in the list are database views instead of tables.* 306
- FIGURE 7.18:** *Database connection's data source structure showing the added Customers and Orders tables* 307
- FIGURE 7.19:** *Adding the Orders child table to the Customers parent table* 307
- FIGURE 7.20:** *Defining a relationship between the child and parent tables. Some relationships are automatically defined for you.* 308
- FIGURE 7.21:** *Sort Order dialog for setting three levels of sorting order* 310

- FIGURE 7.22:** *Edit SQL dialog, which allows hand-editing of the query statement used by InfoPath to create the data source* 311
- FIGURE 7.23:** *Customers secondary data source* 313
- FIGURE 7.24:** *A nested database table (Orders), which is hard to represent when the parent table (Customers) is in the view as a Repeating Table* 315
- FIGURE 7.25:** *Our rendition of displaying the Customers and Orders data* 316
- FIGURE 7.26:** *Preview of the modified form layout with Northwind customers and orders data* 316
- FIGURE 7.27:** *Navigating to the Web service* 319
- FIGURE 7.28:** *Browser test form for the SaveItem Web service method* 320
- FIGURE 7.29:** *Entering an address to the WSDL* 321
- FIGURE 7.30:** *List of methods from the Web service that can be used to receive data* 323
- FIGURE 7.31:** *Setting up the parameter to the SaveItems Web service method, which involves selecting which data source node provides its value* 324
- FIGURE 7.32:** *Choosing the data source node to submit to the SaveItems method. The data source is based on the query Web service data connection.* 325
- FIGURE 7.33:** *InfoPath Designer after finishing the Web service Data Connection Wizard* 326
- FIGURE 7.34:** *Querying and submitting our Web service form during preview* 327
- FIGURE 7.35:** *The Rule Details dialog for changing a REST URL* 330
- FIGURE 7.36:** *Designing the TerraServer form using the GetTile Web service method* 331
- FIGURE 7.37:** *Querying the TerraServer for a section of the San Francisco Bay near Highway 80* 332
- FIGURE 7.38:** *Searching for Web services by using UDDI in InfoPath* 334
- FIGURE 7.39:** *Converting an InfoPath data connection to use a server-defined data connection file* 336
- FIGURE 7.40:** *Secondary data source-bound controls allow for little customization* 339
- FIGURE 7.41:** *Form after querying the XML file secondary data connection* 341



- FIGURE 7.42:** *Enabling query support for offline mode when adding or modifying a secondary data connection* 344
- FIGURE 7.43:** *Configuring a form template for offline mode queries* 345
- FIGURE 7.44:** *InfoPath global cached queries setting* 346
- FIGURE 7.45:** *Error dialog shown if querying an external data source fails when filling out the form* 347
- FIGURE 7.46:** *Error dialog that results from clicking Try to Connect in Figure 7.45 if the data connection is unavailable* 348
- FIGURE 7.47:** *Grocery list form template, unchanged, from Chapter 5* 350
- FIGURE 7.48:** *Drop-Down List Box Properties dialog* 351
- FIGURE 7.49:** *Selecting a secondary data source node from the XML file connection that provides data for the List Box entries* 352
- FIGURE 7.50:** *Filling out the grocery list form when the items are retrieved from a data connection* 353
- FIGURE 7.51:** *Using the main data source to provide data for a List Box control* 354
- FIGURE 7.52:** *Adding a note-taking capability to our grocery form* 355
- FIGURE 7.53:** *Filling out the form after we've added the note-taking feature* 356
- FIGURE 7.54:** *Specifying a filter to show only grocery items from a selected category* 357
- FIGURE 7.55:** *Relationships between a form and its external data sources* 358
-
- FIGURE 8.1:** *Settings for enabling and disabling certain features for users, such as the ability to save a form* 364
- FIGURE 8.2:** *The first view of the job application form* 365
- FIGURE 8.3:** *The last page of the job application form* 366
- FIGURE 8.4:** *Dialog about the submit error that occurs when the user clicks the Submit button after not filling in a name in the first view* 367
- FIGURE 8.5:** *Submit Options dialog, the starting point for enabling submit* 368
- FIGURE 8.6:** *A custom list in SharePoint* 371
- FIGURE 8.7:** *The default list entry form* 372
- FIGURE 8.8:** *The Customize Form button* 372
- FIGURE 8.9:** *Field properties dialog for a list field* 373
- FIGURE 8.10:** *Configuring SharePoint document library submit for the student sign-in form* 375

- FIGURE 8.11:** *Submit Options dialog configured to submit to a SharePoint document library* 376
- FIGURE 8.12:** *SharePoint document library that contains submitted form data* 378
- FIGURE 8.13:** *First page of the Data Connection Wizard for setting properties of the e-mail sent when users submit the form* 379
- FIGURE 8.14:** *Configuring the attachment options for submitting via e-mail* 380
- FIGURE 8.15:** *Dialog prompt in which the user must click Send in order to submit a helpdesk request* 381
- FIGURE 8.16:** *Specifying whether your form only receives data or only submits data with a Web service* 386
- FIGURE 8.17:** *Selecting a Web service method for submit* 387
- FIGURE 8.18:** *InfoPath Designer after finishing the Data Connection Wizard for submitting data to a Web service* 388
- FIGURE 8.19:** *Error dialog that results when submitting a form with validation errors that are in the current view* 389
- FIGURE 8.20:** *Student sign-in form during preview* 390
- FIGURE 8.21:** *Dialog confirming that submit succeeded* 390
- FIGURE 8.22:** *Student sign-in form showing the instructor's Class Information view with default data* 392
- FIGURE 8.23:** *Designing the sign-in form view* 394
- FIGURE 8.24:** *Configuring conditional formatting to hide the LateReason section if the start time is later than now* 394
- FIGURE 8.25:** *Setting custom data validation to make the Reason field required when the LateReason section is visible* 396
- FIGURE 8.26:** *Defining parameter mappings for Web service submit* 397
- FIGURE 8.27:** *Tardy Johnny signs in to Philosophy 101* 398
- FIGURE 8.28:** *Debugging the Web service by using Visual Studio* 400
- FIGURE 8.29:** *Debugger paused at our breakpoint* 400
- FIGURE 8.30:** *Data source for the student sign-in form* 402
- FIGURE 8.31:** *Parameter mappings dialog configured to submit the entire form* 403
- FIGURE 8.32:** *Dialog that gives the user a chance to digitally sign a form before continuing with submit* 408
- FIGURE 8.33:** *Controlling whether or not to display the prompt shown in Figure 8.32* 408
- FIGURE 8.34:** *Submit Options dialog for Web server submit via HTTP* 409



- FIGURE 8.35:** *Setting the destination option to use an ASP.NET page* 410
- FIGURE 8.36:** *Submit Options Dialog, choosing to run rules* 413
- FIGURE 8.37:** *List of rules that run when the form is submitted* 414
- FIGURE 8.38:** *Actions for the actual submit to the Web services* 415
- FIGURE 8.39:** *Expression shown in a popup message when a student signs in late to class* 415
- FIGURE 8.40:** *Dialog that appears when Johnny signs in tardy* 416
- FIGURE 8.41:** *Dialog that appears when submit with rules fails on a secondary connection* 418
- FIGURE 8.42:** *Dialog shown when submit using rules fails on the main submit connection* 419
- FIGURE 8.43:** *Generic success dialog for the main submit connection* 420
- FIGURE 8.44:** *Generic failure dialog for the main submit connection* 421
- FIGURE 8.45:** *Advanced section of the Submit Options dialog* 421
-
- FIGURE 9.1:** *InfoPath prompt that appears when choosing Save or Save As for a new form template* 427
- FIGURE 9.2:** *The Publish menu, offering a variety of publishing options* 428
- FIGURE 9.3:** *Publishing a MeetingRequest template to a Web server via a “backdoor” network share* 429
- FIGURE 9.4:** *Specifying the users’ access path* 431
- FIGURE 9.5:** *Error dialog that appears when opening a form from a path that is not the access path* 431
- FIGURE 9.6:** *Specifying a server location for publishing to a SharePoint site or Forms Services site* 434
- FIGURE 9.7:** *Choosing whether to publish the form template to a SharePoint document library or a site content type* 435
- FIGURE 9.8:** *Saving the template somewhere on the SharePoint server when creating a site content type* 436
- FIGURE 9.9:** *Wizard page showing promoted properties from the template’s data source, which appear as columns in SharePoint libraries and lists* 437
- FIGURE 9.10:** *Form library on SharePoint* 437
- FIGURE 9.11:** *Adding site content types to a document library* 438
- FIGURE 9.12:** *Selecting a field from the data source to promote. The display name (Due Date) can be set independently from the node name (when).* 440

- FIGURE 9.13:** *Choosing to publish to Forms Services via the Document Library option* 443
- FIGURE 9.14:** *Last page of the Publishing Wizard before clicking the Publish button* 444
- FIGURE 9.15:** *Last page of the Publishing Wizard after clicking the Publish button* 445
- FIGURE 9.16:** *First wizard page for publishing a form to e-mail recipients* 446
- FIGURE 9.17:** *Promoting properties to make them available as columns in Outlook* 447
- FIGURE 9.18:** *Resulting e-mail in Outlook after finishing the Publishing Wizard for e-mail* 448
- FIGURE 9.19:** *Browse For Folder dialog for selecting a location to save template source files* 452
- FIGURE 9.20:** *Dialog warning that appears when saving source files to a folder that already contains existing source files* 452
- FIGURE 9.21:** *Configuring preview settings* 455
- FIGURE 9.22:** *Error message that results from attempting to open a form that was saved while previewing* 459
-
- FIGURE 10.1:** *Add a Workflow page* 465
- FIGURE 10.2:** *Customizing the Travel Request Approval workflow* 466
- FIGURE 10.3:** *Workflow initiation e-mail message* 467
- FIGURE 10.4:** *E-mail task requesting approval of the travel request* 467
- FIGURE 10.5:** *Travel Request form showing the Workflow Task message bar* 468
- FIGURE 10.6:** *Workflow dialog* 469
- FIGURE 10.7:** *Workflows dialog* 471
- FIGURE 10.8:** *The Open SharePoint Site interface* 472
- FIGURE 10.9:** *The navigation menu of SharePoint Designer 2010* 472
- FIGURE 10.10:** *The Create Reusable Workflow dialog for Data Collection* 473
- FIGURE 10.11:** *The Action drop-down list* 474
- FIGURE 10.12:** *The inserted action* 474
- FIGURE 10.13:** *Select Users dialog* 475
- FIGURE 10.14:** *Custom Task Wizard, form fields* 476
- FIGURE 10.15:** *Add Field dialog* 476
- FIGURE 10.16:** *Association and Initiation Form Parameters dialog* 477



- FIGURE 10.17:** *Settings navigation* 478
- FIGURE 10.18:** *Manage User Roles dialog* 481
- FIGURE 10.19:** *Add User Role dialog* 482
- FIGURE 10.20:** *Select Users dialog* 483
- FIGURE 10.21:** *Manage User Roles dialog after adding a few roles* 484
- FIGURE 10.22:** *The Form Load button* 487
- FIGURE 10.23:** *Selecting User's current role as the qualifier in the Condition dialog for rules* 487
- FIGURE 10.24:** *Rules pane after setting the condition and action* 488
- FIGURE 10.25:** *Rules for Opening Forms dialog after adding all rules* 489
- FIGURE 10.26:** *Preview tab on the Form Options dialog* 490
- FIGURE 10.27:** *Selecting the UserRole field that corresponds to the TeamName field in the secondary data source* 493
- FIGURE 10.28:** *Insert Formula dialog after adding a calculated default value* 493
- FIGURE 10.29:** *Specifying a user role in the condition for data validation* 495
- FIGURE 10.30:** *Rules Pane after changing the condition to use a user role* 496
- FIGURE 10.31:** *Manage User Roles dialog after adding "Team Manager" and "Sales Manager" roles* 497
- FIGURE 10.32:** *Rules Pane after adding a rule to switch to the Team Status Report view* 498
- FIGURE 10.33:** *Food survey form template* 503
- FIGURE 10.34:** *Choosing which properties to make available in Outlook folders* 503
- FIGURE 10.35:** *Sending the form to a list of e-mail recipients* 504
- FIGURE 10.36:** *Rules Wizard showing the rule specific to InfoPath forms* 505
- FIGURE 10.37:** *Choose InfoPath Form dialog* 506
- FIGURE 10.38:** *Create New Folder dialog after selecting InfoPath Form Items* 508
- FIGURE 10.39:** *Folder properties dialog after clicking on the InfoPath Forms tab* 509
- FIGURE 10.40:** *Team Lunch folder showing "Food Preference" column* 509
- FIGURE 10.41:** *InfoPath e-mail form in Outlook's reading pane* 510
- FIGURE 10.42:** *Filling out an InfoPath e-mail form* 511
- FIGURE 10.43:** *Submit dialog when submitting an InfoPath e-mail form* 512
- FIGURE 10.44:** *Mail Options task pane* 512
- FIGURE 10.45:** *Team Lunch folder showing responses to the food preference survey* 513
- FIGURE 10.46:** *Team Lunch folder grouped by food preference* 514

- FIGURE 10.47:** *Expense Reports InfoPath form folder* 515
- FIGURE 10.48:** *InfoPath actions context menu* 516
- FIGURE 10.49:** *E-Mail Attachments tab in the Form Options dialog* 517
-
- FIGURE 11.1:** *Configuring security and trust settings for a form template* 527
- FIGURE 11.2:** *Form Template Properties dialog for setting the form ID* 528
- FIGURE 11.3:** *Error dialog when opening a restricted form template that attempts to query an external data source* 530
- FIGURE 11.4:** *Setting an access path, which is not required when publishing a restricted form template* 531
- FIGURE 11.5:** *Cache conflict that occurs when a different version of the same form template is opened* 532
- FIGURE 11.6:** *New dialog showing form templates in the InfoPath cache* 534
- FIGURE 11.7:** *Resource Files dialog, which shows files accessible by all form templates* 534
- FIGURE 11.8:** *Internet Explorer Web content zones, each of which has its own security level* 537
- FIGURE 11.9:** *Settings for defining the local intranet zone* 538
- FIGURE 11.10:** *Security prompt for cross-domain data access* 542
- FIGURE 11.11:** *Office Customization Tool security settings for InfoPath* 544
- FIGURE 11.12:** *Error dialog that appears when a domain form is not opened from its access path and the published template is unavailable* 547
- FIGURE 11.13:** *Opening a domain form whose access path doesn't match the .xsn location, but the form template exists at the originally published location* 548
- FIGURE 11.14:** *Entering a domain for the template when previewing the form* 549
- FIGURE 11.15:** *List of any settings or features resulting in the form template being at domain trust level* 551
- FIGURE 11.16:** *Security and Trust tab of the Form Options dialog* 553
- FIGURE 11.17:** *Trusted root certification authorities for the current user* 556
- FIGURE 11.18:** *Security and Trust category settings for signing a form template* 557
- FIGURE 11.19:** *Selecting a certificate to use for digital signing* 557
- FIGURE 11.20:** *Creating a self-signed certificate for code signing* 558
- FIGURE 11.21:** *General tab of the Certificate dialog, which shows whether the publisher of a certificate is trusted* 559

- FIGURE 11.22:** *InfoPath prompt that appears when a signed template's CA isn't trusted* 560
- FIGURE 11.23:** *Error that occurs when opening a full trust form template that is neither digitally signed nor installed* 561
- FIGURE 11.24:** *Trusted Publishers category of the Trust Center dialog* 567
- FIGURE 11.25:** *Add-ins category of the Trust Center dialog* 569
- FIGURE 11.26:** *The External Content Category of the Trust Center* 571
- FIGURE 11.27:** *Privacy Options category of the Trust Center dialog* 573
- FIGURE 11.28:** *Configuring digital signatures for a form template* 577
- FIGURE 11.29:** *Our sample MOI Consulting performance review form* 578
- FIGURE 11.30:** *Setting up the template to allow signing the entire form* 579
- FIGURE 11.31:** *Filling out the MOI performance review before using a digital signature* 580
- FIGURE 11.32:** *Digital Signatures dialog, which tracks who has signed this form* 581
- FIGURE 11.33:** *Selecting what form data to sign* 582
- FIGURE 11.34:** *Clicking the Sign button to apply a digital signature to the form* 582
- FIGURE 11.35:** *Additional Information dialog, which shows exactly what you're signing before you sign it* 583
- FIGURE 11.36:** *Digitally signed form with read-only controls* 585
- FIGURE 11.37:** *Error dialog that appears when trying to change signed form data* 585
- FIGURE 11.38:** *Friendly reminder that the form being opened is digitally signed* 586
- FIGURE 11.39:** *Choosing whether or not to show a notification when signed forms are opened* 587
- FIGURE 11.40:** *Setting up the Personal Information part of the view for partial signing* 590
- FIGURE 11.41:** *Creating a new set of signable data (partial signing) on a Section control* 591
- FIGURE 11.42:** *Setting up digital signatures on a Section control* 593
- FIGURE 11.43:** *Signing the "Info" set of signable data when filling out the form* 594
- FIGURE 11.44:** *Using the Select the Data to Sign dialog opened by clicking the Digital Signatures button* 594
- FIGURE 11.45:** *Signing a set of signable data when filling out the form* 595
- FIGURE 11.46:** *Form showing that the set of signable data has been signed* 596
- FIGURE 11.47:** *Signature Details dialog for a signed Section control* 597
- FIGURE 11.48:** *Form showing that Robert co-signed John's review rating* 599

- FIGURE 11.49:** *The Sign dialog from the Signature Line control* 606
- FIGURE 11.50:** *Specifying permission in InfoPath Designer to apply when the form is filled out* 608
- FIGURE 11.51:** *Specifying permission for the current form while filling it out* 610
- FIGURE 11.52:** *Form Permission task pane* 611
- FIGURE 11.53:** *Permission dialog after clicking the More Options button* 612
- FIGURE 11.54:** *Enabling Information Rights Management on the SharePoint Central Administration site* 615
- FIGURE 11.55:** *IRM settings for a document library* 616
-
- FIGURE 12.1:** *Weekly status report form* 621
- FIGURE 12.2:** *Merge Forms dialog* 622
- FIGURE 12.3:** *Merge Forms dialog showing the Views menu* 623
- FIGURE 12.4:** *Weekly status report document library in Merge Forms view* 624
- FIGURE 12.5:** *Merge error dialog for forms with a schema that doesn't match the target form* 624
- FIGURE 12.6:** *Merged status report form* 626
- FIGURE 12.7:** *Data source for the status report form template* 628
- FIGURE 12.8:** *Weekly Status Report view in the weekly status report form template* 629
- FIGURE 12.9:** *Team Status Report view for merging multiple status report forms* 630
- FIGURE 12.10:** *Merge tab on the Field or Group Properties dialog* 632
- FIGURE 12.11:** *Advanced tab of the Repeating Section Properties dialog* 633
- FIGURE 12.12:** *Merge settings error dialog* 634
- FIGURE 12.13:** *Merge Settings dialog for the StatusReports repeating group node* 635
- FIGURE 12.14:** *Merge Settings dialog with merge customizations* 637
- FIGURE 12.15:** *Merge Settings dialog for the EmployeeInformation node* 639
- FIGURE 12.16:** *Team status report after merging three status reports* 640
- FIGURE 12.17:** *Merge Settings dialog for the Summary node* 641
- FIGURE 12.18:** *Merge Settings dialog for the Summary node after customizations are complete* 643
- FIGURE 12.19:** *Merge Settings dialog for the ThisWeek node after customization* 643



FIGURE 12.20: *Merged team status report after all customizations are complete* 644

FIGURE 12.21: *Advanced tab of the Form Options dialog* 645

FIGURE 12.22: *Field or Group Properties dialog after specifying a custom merge XSL in the manifest* 648

FIGURE 12.23: *Create New View dialog* 653

FIGURE 12.24: *Printed team status report* 653

FIGURE 12.25: *Print Settings tab of the View Properties dialog* 654

FIGURE 12.26: *Header format dialog* 656

FIGURE 12.27: *Header format dialog after adding Team field and date* 657

FIGURE 12.28: *Team status report with header and footer information* 658

FIGURE 12.29: *Sales Report view* 659

FIGURE 12.30: *Merged sales report data* 659

FIGURE 12.31: *Print dialog* 660

FIGURE 12.32: *Print Multiple Views dialog when filling out a form* 660

FIGURE 12.33: *Print Multiple Views dialog in InfoPath Designer* 662

FIGURE 12.34: *XML Structure task pane in Microsoft Office Word* 664

FIGURE 12.35: *Word document used to generate the Word print view XSL* 665

FIGURE 12.36: *Word Print Views dialog* 666

FIGURE 12.37: *Choosing the path and file name from the Add Print View for Word Wizard* 667

FIGURE 12.38: *Specifying the Word print view name in the Add Print View for Word Wizard* 667

FIGURE 12.39: *Filling out the Team Status Report view in the browser* 669

FIGURE 12.40: *Print View button on the Forms Services toolbar* 669

FIGURE 12.41: *Team Status Report view shown as a print view in the browser* 670

FIGURE 12.42: *Export To in the File tab* 671

FIGURE 12.43: *Choosing the type of data to export* 672

FIGURE 12.44: *Choosing the forms from which to export data* 672

FIGURE 12.45: *Status report data exported to Microsoft Office Excel* 673

FIGURE 12.46: *Export to Excel Wizard after choosing to export only the data from a specific table* 674

FIGURE 12.47: *Selecting the data to export* 674

FIGURE 12.48: *Sales data exported to Microsoft Office Excel* 675

FIGURE 12.49: *Filler Features tab of the Form Options dialog* 676

- FIGURE 13.1:** *Programming category in the Form Options dialog* 682
- FIGURE 13.2:** *Confirmation dialog shown before removing code* 682
- FIGURE 13.3:** *Upgrading code from a form template compatible with InfoPath 2003* 683
- FIGURE 13.4:** *Programming options, which are saved as defaults for all new form templates* 686
- FIGURE 13.5:** *VSTA development environment* 687
- FIGURE 13.6:** *EventBubbling sample, which shows Site and Sender node names as events bubble up the data source* 695
- FIGURE 13.7:** *MOI Consulting morale event scheduler form in InfoPath Designer* 700
- FIGURE 13.8:** *Password dialog prompt* 701
- FIGURE 13.9:** *Custom error message that appears when a Changing event handler is canceled* 705
- FIGURE 13.10:** *Error that occurs when calling unsupported OM during the Changing event* 707
- FIGURE 13.11:** *Show Error Message dialog* 717
- FIGURE 13.12:** *Supporting entry of last and first names in a Repeating Table control* 719
- FIGURE 13.13:** *MultipleNotifications sample form in InfoPath Designer* 725
- FIGURE 13.14:** *One of the notifications when adding underline to a rich text (XHTML) field* 726
- FIGURE 13.15:** *Setting up a custom task pane in the Form Options dialog* 736
- FIGURE 13.16:** *Welcome view of the MOI Consulting request form* 740
- FIGURE 13.17:** *Dialog that results when the Time-sensitive/Critical request type is selected* 741
- FIGURE 13.18:** *Dialog that appears when form errors exist in the current view* 742
- FIGURE 13.19:** *Informing the user that signing is required to continue* 742
- FIGURE 13.20:** *Dialog that appears after using the Click here to sign this section link in the MOI request form* 743
- FIGURE 13.21:** *Request Details view of the MOI Consulting request form* 744
- FIGURE 13.22:** *Confirm view of the MOI Consulting request form* 745
- FIGURE 13.23:** *Thank You view of the MOI Consulting request form* 745
- FIGURE 13.24:** *Main data source for the MOI Consulting request form template* 750

FIGURE 13.25: *Error dialog that appears when user attempts to open an existing MOI request form* 756

FIGURE 13.26: *Dialog for clearly presenting to the user any form errors* 760

FIGURE 13.27: *GetSubAreas secondary data source* 770

FIGURE 13.28: *Request Details view* 775

FIGURE 13.29: *Finding the xmlToEdit and ActionType capability of a structurally editable control* 777

FIGURE 13.30: *Setting up the form's main submit to use code* 779

FIGURE 13.31: *Our own message box* 784

FIGURE 14.1: *Design Checker messages from the MOI feedback form template* 793

FIGURE 14.2: *Controls targeted by the messages in the Design Checker for the MOI feedback form template* 794

FIGURE 14.3: *Postback settings for a control in a browser-enabled form template* 797

FIGURE 14.4: *Customizing user interface buttons for a browser-enabled form template* 798

FIGURE 14.5: *Configuring a Button control to use the Update Form option for a browser-enabled form template* 799

FIGURE 14.6: *Accessibility control settings* 800

FIGURE 14.7: *Publishing to Forms Services a form template that requires administrator approval* 804

FIGURE 14.8: *Specifying a path during publishing to save a form template requiring administrator approval* 805

FIGURE 14.9: *Administration links for Forms Services* 807

FIGURE 14.10: *Upload Form Template page* 808

FIGURE 14.11: *Manage Form Templates page* 809

FIGURE 14.12: *Choosing a site collection and Web application for activation* 811

FIGURE 14.13: *Activating a form template on the Site Collection Features page* 812

FIGURE 14.14: *Forms Services Admin tool authored by using the admin object model* 816

FIGURE 14.15: *Some settings for configuring InfoPath Forms Services* 821

FIGURE 14.16: *Configuring the session state shared service* 823

FIGURE 14.17: *Administrative session state thresholds* 825

FIGURE 14.18: *Quiescing error that frustrated John when creating a status report form* 829

- FIGURE 14.19:** *Configuring to quiesce a form template on the Quiesce Form Template page* 829
- FIGURE 14.20:** *Status of a quiescing form template* 829
- FIGURE 14.21:** *Fully quiesced form template* 830
- FIGURE 14.22:** *Quiesce tool authored by using admin object models* 834
- FIGURE 14.23:** *Diagnostic Logging administration page* 840
- FIGURE 14.24:** *Global administrative data connection settings for Forms Services* 845
- FIGURE 14.25:** *The Convert Data Connection dialog* 852
- FIGURE 14.26:** *Connection Options dialog when modifying a server-based connection* 852
- FIGURE 14.27:** *Uploading to the centrally managed connection library* 853
- FIGURE 14.28:** *Passing NTLM credentials for data connections with InfoPath and Forms Services* 857
- FIGURE 14.29:** *Settings on the Managing the Web Service Proxy page* 860
- FIGURE 14.30:** *Copying the physical path for the SharePoint root site collection from IIS Manager* 866
- FIGURE 14.31:** *Opening the SharePoint site by using the file system path* 867
- FIGURE 14.32:** *Solution Explorer after adding the XmlFormView folder and MyPage.aspx Web form* 868
- FIGURE 14.33:** *Toolbox after adding the XmlFormView control* 869
- FIGURE 14.34:** *Hosting the life insurance form in MyPage.aspx* 872
- FIGURE 14.35:** *Adding the Initialize event handler in the host page for the XmlFormView1 control* 876
- FIGURE 14.36:** *Host page handling the form submit by sending a custom response* 883
- FIGURE 14.37:** *Clicking the Validate Form button shows a dialog box message in the browser* 886
- FIGURE 14.38:** *State of the life insurance form while the dialog box still has focus* 887
- FIGURE 14.39:** *Configuring Forms Services form session state* 895
- FIGURE 14.40:** *Adding performance counters for templates activated with InfoPath Forms Services* 897
- FIGURE 14.41:** *SharePoint 2010 Management Console* 900
-
- FIGURE 15.1:** *Document properties dialog in Word 2003* 906
- FIGURE 15.2:** *Document Information Panel in Word 2010* 907



- FIGURE 15.3:** *Document Information Panel dialog* 907
- FIGURE 15.4:** *SharePoint Document Properties page* 908
- FIGURE 15.5:** *Document Information Panel Settings page* 909
- FIGURE 15.6:** *List of content types in the Data Source Wizard* 910
- FIGURE 15.7:** *InfoPath form template based on the Document content type after adding the Assigned To column in SharePoint* 911
- FIGURE 15.8:** *Data source for the Document content type* 912
- FIGURE 15.9:** *Customized Document Properties view for editing standard document properties* 913
- FIGURE 15.10:** *Completing the wizard will publish the Document Information Panel template back to the SharePoint document library* 914
- FIGURE 15.11:** *Document Information Panel showing the Document Properties—Server view* 914
- FIGURE 15.12:** *Document Information Panel showing the Document Properties view* 915
- FIGURE 15.13:** *Word document showing the Document Information Panel and the Author property in the body of the document* 915
- FIGURE 15.14:** *Visual Studio Toolbox after adding the InfoPath FormControl control* 918
- FIGURE 15.15:** *IP Insurance application after adding Combo Box and GroupBox controls* 923
- FIGURE 15.16:** *IP Insurance application after loading the auto insurance form* 925
- FIGURE 15.17:** *Standard toolbar for the IP Insurance application* 929
- FIGURE 15.18:** *Submit Options dialog with Hosting environment as the destination* 947
- FIGURE 15.19:** *Add Custom Control Wizard showing the Microsoft UpDown Control 6.0 (SP4) selected in the list of ActiveX controls* 951
- FIGURE 15.20:** *Specify Installation Options page of the Add Custom Control Wizard* 952
- FIGURE 15.21:** *Specify a Binding Property page of the Add Custom Control Wizard* 953
- FIGURE 15.22:** *Specify an Enable or Disable Property page of the Add Custom Control Wizard* 955
- FIGURE 15.23:** *Specify Data Type Options page of the Add Custom Control Wizard* 956

- FIGURE 15.24:** *Specify Data Type Options page showing the Field (element with custom data type) option* 958
- FIGURE 15.25:** *Properties dialog for the Microsoft UpDown Control* 960
-
- FIGURE 16.1:** *MOI Consulting expense report form created in Word* 966
- FIGURE 16.2:** *Import Wizard page for selecting the importer to use* 968
- FIGURE 16.3:** *Import Wizard page for selecting the file to import* 968
- FIGURE 16.4:** *Import Options dialog for importing Word documents* 969
- FIGURE 16.5:** *Import Options dialog for importing Excel workbooks* 970
- FIGURE 16.6:** *Final page of the Import Wizard* 971
- FIGURE 16.7:** *Expense report form after importing into InfoPath* 972
- FIGURE 16.8:** *Change To Repeating Table dialog* 974
- FIGURE 16.9:** *Registry key structure for importers and exporters* 977
- FIGURE 16.10:** *Export Wizard showing a custom exporter* 979
- FIGURE 16.11:** *Importer/exporter process flow* 981
-
- FIGURE A.1:** *SharePoint Designer List details Forms section* 995
- FIGURE A.2:** *SharePoint List Ribbon Customize List section* 999
- FIGURE A.3:** *Updated InfoPath form* 1004
- FIGURE A.4:** *List View Web Parts* 1007
- FIGURE A.5:** *Create page* 1008
- FIGURE A.6:** *Tactical Projects Value Calculation section* 1009
- FIGURE A.7:** *Creating a New Mashup Wiki page* 1011
- FIGURE A.8:** *InfoPath Form Web Part and Projects List View Web Part* 1013
- FIGURE A.9:** *All Web Parts added to mashup page* 1014
- FIGURE A.10:** *Project Statuses, Type, and Status Web Parts configured* 1016
- FIGURE A.11:** *All Web Parts of mashup page configured* 1018



Tables

- TABLE 2.1:** *Starting Points for Form Design* 31
- TABLE 2.2:** *Data Requirements for the MOI Consulting Employment Application* 34
- TABLE 2.3:** *Input Controls* 44
-
- TABLE 3.1:** *Controls Supported in Forms Services* 91
- TABLE 3.2:** *Matrix of Browsers Supported for Use with Forms Services* 94
- TABLE 3.3:** *Form Features that May Automatically Communicate with the Server* 106
-
- TABLE 4.1:** *Icons Shown in the Fields Task Pane for Various Types of Data Source Nodes* 113
- TABLE 4.2:** *InfoPath 2010 Controls and Their Binding Behaviors* 114
- TABLE 4.3:** *XML Data and Schema that Match Before Changing a Field or Group Name* 122
- TABLE 4.4:** *Mismatched XML Data and Schema after a Name Change* 123
- TABLE 4.5:** *How Data Source Operations on a Published Form Template May Affect Saved Forms* 123
- TABLE 4.6:** *Data Types Available in InfoPath Designer and Their Value Ranges* 125
- TABLE 4.7:** *Sample Form XML Data with and without a Repeating Container** 131

- TABLE 4.8:** *Data from Table 4.7, but with Updated Node Names for Readability** 131
- TABLE 4.9:** *InfoPath Controls and the XML Schema Created “Behind” the Data Source in the XSD File* 137
- TABLE 4.10:** *Designing a Music Collection Form’s Data Source with and without Attributes* 147
- TABLE 4.11:** *Analyzing Nodes’ Namespaces Relative to Their Immediate Ancestors* 165
- TABLE 4.12:** *Control, Node, and Other Unrelated Properties that Change or Get Reset after Rebinding* 171
- TABLE 4.13:** *The Behaviors of Various Controls when Multiply Bound to field1 while Filling Out the Form* 178
- TABLE 4.14:** *Design-Time Visual Icons that Appear Over Controls to Indicate Potential Binding Issues* 178
- TABLE 4.15:** *When Design-Time Visuals Appear on Controls* 179
-
- TABLE 5.1:** *Controls that Support Input Scope* 229
-
- TABLE 6.1:** *Actions Available for a Rule* 268
-
- TABLE 7.1:** *Types of Main and Secondary Data Query Connections* 287
- TABLE 7.2:** *Descriptions of the Web Service Methods Used* 319
- TABLE 7.3:** *Available and Unavailable Features for Secondary Data Sources* 341
-
- TABLE 8.1:** *Unsupported Database Data Types for Submit* 384
- TABLE 8.2:** *Matrix of Submit Possibilities among Fields, Groups, and the Include Setting* 402
-
- TABLE 9.1:** *Persistence of Publishing Options in the Form Definition (.xsd) File* 450
- TABLE 9.2:** *Disabled Commands when Previewing a Form* 454
-
- TABLE 10.1:** *Workflows Included with Microsoft SharePoint Server 2010* 463

- TABLE 11.1:** *OM Security-Level Definitions* 540
- TABLE 11.2:** *Form and IE Security-Level Impacts on OM Security Level* 541
- TABLE 11.3:** *CAS Permission Set for LocalIntranet* 563
- TABLE 11.4:** *Mapping of Default SharePoint Roles to IRM Permissions* 617
- TABLE 11.5:** *Direct Mapping of SharePoint Permissions to IRM* 618
-
- TABLE 12.1:** *Default Merge Actions Based on Node Type* 627
- TABLE 12.2:** *Merge Customizations Available for Each Node Type* 635
- TABLE 12.3:** *List of Available Separators in the Merge Settings Dialog for Rich Text Fields* 641
- TABLE 12.4:** *Merge Actions in the Aggregation Namespace* 650
- TABLE 12.5:** *Merge Attributes Available for Merge Actions* 650
-
- TABLE 13.1:** *Programming Languages and Available InfoPath Object Model Versions* 681
- TABLE 13.2:** *Form Events Exposed by InfoPath* 692
- TABLE 13.3:** *Invalid Object Model Calls During XML Events* 706
- TABLE 13.4:** *Examples of Advanced XPath Paths for Registering XML Event Handlers* 733
- TABLE 13.5:** *Types of Data Connections and Their Commonly Used Properties and Methods* 773
- TABLE 13.6:** *IsBrowser and IsMobile Values in Various Environments* 787
- TABLE 13.7:** *Object Model Classes, Events, Properties, and Methods Not Implemented by Forms Services* 788
-
- TABLE 14.1:** *Full List of stsadm.exe Commands Specific to Forms Services* 813
- TABLE 14.2:** *New and Existing Form-Filling Sessions During Quiescing and Quiesced Status* 830
- TABLE 14.3:** *Commands for stsadm.exe Specific to Quiescing* 833
- TABLE 14.4:** *Administration OM on the FormTemplate Object Specific to Quiescing* 833
- TABLE 14.5:** *Forms Services Errors that May Appear in the Windows Event Log* 836
- TABLE 14.6:** *Forms Services Warnings that May Appear in the Windows Event Log* 838

- TABLE 14.7:** *Commonly Used Properties of the XmlFormView Control* 870
- TABLE 14.8:** *All Properties of the XmlFormView Control (Includes Properties from Table 14.7 but No Inherited Properties)* 879
- TABLE 14.9:** *Properties and Methods Not Available to the ASP.NET Host Page* 880
- TABLE 14.10:** *Events the ASP.NET Host Page May Not Sink* 881
- TABLE 14.11:** *Complete List of InfoPath Forms Services Performance Counters* 897
-
- TABLE 15.1:** *InfoPath Form Control Methods* 920
- TABLE 15.2:** *InfoPath Form Control Properties* 922
- TABLE 15.3:** *Field or Group Types* 957
- TABLE 15.4:** *Interfaces and Properties that Controls Should Implement to Work Well in InfoPath* 961
-
- TABLE 16.1:** *String Values to Be Added Under Each LCID Key* 978
- TABLE 16.2:** *Methods of the IFormTemplateConverter2 Interface* 981
- TABLE 16.3:** *Methods of the IConversionManager Interface* 982
- TABLE 16.4:** *Method of the IConversionManager2 Interface* 982
- TABLE 16.5:** *Parameters for the Import and Export Methods* 984



Foreword

Designing Forms for SharePoint and InfoPath is a hands-on introduction to Microsoft InfoPath.

Like the Web, InfoPath continues to grow and change and has evolved toward a holistic story around rapid development of workflow applications on Microsoft Office SharePoint Server. Together with Microsoft SharePoint Designer 2010, InfoPath 2010 facilitates creation of end-to-end solutions that feature powerful forms together with enterprise-scale workflow and access to key business data. InfoPath was designed, at its core, as a powerful XML editing engine that enables end users to interface easily with data.

The present book is based largely on *Designing Forms for Microsoft Office InfoPath and Forms Services 2007*, which was written by two distinguished members of the InfoPath product team who designed, implemented, and tested many of the core features of the product. In revising the book for this edition, we have tried to keep the deep encyclopedic knowledge of InfoPath embodied in that work, while also restructuring the material to focus attention on the SharePoint application development experience. The book has been updated as well to call attention to the new features introduced in InfoPath 2010 to help build more powerful SharePoint applications—features such as:

- Customizing the forms used to create, view, and edit SharePoint list items



- Dynamic queries to REST Web services
- The InfoPath Form Web part, which allows you to create powerful Web parts without writing code, and to connect them with other Web parts to create data mashups

What's interesting and unique about InfoPath is the type of information it allows people to gather. InfoPath lets organizations design and edit "semistructured" documents, or documents that have regions of meaning, in the same way that columns in a database have meaning. While the program provides great design and editing capabilities for traditional forms such as purchase orders and equipment requests, InfoPath innovatively yet squarely targets information that historically has been more difficult to capture, such as business-critical data contained in sales reports, inventory updates, project memos, travel itineraries, and performance reviews. InfoPath was born as a tool for editing XML, and XML is about creating documents in which the content is delimited, or set apart, by tags that explain the meaning of each piece of content. With XML, documents can become a source of information as rich as a database, enabling search, processing, and reuse. The underlying structure of the information in an InfoPath template is described using a **schema**. A schema describes how the data is constructed, in the same way that a blueprint describes how a building is constructed. In SharePoint, these schemas are represented as content types in lists and libraries, and InfoPath provides a consistent way to author forms and logic that turn these lists and libraries into powerful applications that automate processes that previously required many manual steps.

Microsoft's long-term vision for InfoPath has always been about more than the ability to rapidly create forms: It is about building complete end-to-end applications using the power of XML and workflow. Together with the powerful collaboration features of SharePoint, InfoPath is a key part of the toolset you need to rapidly create applications that meet your enterprise application needs. InfoPath 2010 and InfoPath Forms Services in SharePoint 2010 empower business users to automate their own business processes that collect, manage, and share information. IT, developers, and power users can create powerful business applications on the SharePoint



platform using InfoPath forms to interact with external data, to drive workflow, and to enhance Web pages.

—*Jean Paoli*

General Manager, Interoperability and XML Architecture, Microsoft Corporation

Co-creator of the W3C XML 1.0 Recommendation

Co-creator of Microsoft Office InfoPath

—*Nick Dallett*

Program Manager, InfoPath 2003, InfoPath 2007, InfoPath 2010

This page intentionally left blank



Preface

It Just Makes Sense

Over the past decade or so, Extensible Markup Language (XML) has become more widely used than ever before as a means of transferring data between applications and even between organizations. XML provides a standard protocol with which these applications and organizations can communicate. Using XML Schema, a company can define a standard structure for its data that can then be used across multiple departments and organizations. This structured data enables developers to easily create applications that can communicate with each other without much effort.

In addition, most organizations use forms in one way or another, whether to enter a purchase request, submit expense report information, or track weekly status. If you look at a typical form, you will notice that the form itself is structured unlike a typical freeform document created in an application such as Microsoft Office Word 2010. In these freeform documents you can type anything you like in any way that you choose. Although a form may contain sections that allow you to enter freeform text such as comments, most of your typical forms are highly structured. Fields in the form usually require you to enter specific types of data such as sales numbers or costs. Since XML defines a structured data format (which can contain some unstructured elements) and forms are highly structured with



bits of freeform data, it makes sense to tie together forms and XML data. Once a user has filled out a form that is connected to XML data, the data can easily be incorporated into back-end processes that understand the structure of the XML data for that form. So, this fits one of the main purposes of XML—tying together multiple processes using a standard protocol.

Since building forms based on XML just makes sense, many software developers want to create forms-based applications to collect data and store it as XML. However, until a few years ago, this was a tedious and time-consuming process. Developers had to use tools such as Microsoft Visual C++, C#, or Visual Basic .NET and write sometimes a tremendous amount of code to create a forms application. Often, forms applications share similar functionality, such as spell checking, calculations, and data validation. In order to share this functionality across multiple forms applications, software developers needed to create code libraries in order to reuse their code. This worked fine when sharing the code within the same department or company. However, developers across multiple companies were likely going to duplicate the same work unless, of course, companies purchased these libraries from a third-party vendor.

Developing forms applications in this way is not something that typical information workers can do. Usually this type of coding is reserved for advanced software developers. Another disadvantage of this approach is that different forms applications usually have different user interfaces. Each time a user fills out a form, he or she may need to learn a different set of commands and menu items. This learning curve costs the company time and money.

About eight years ago, Microsoft recognized the need for a common tool to build forms based on XML technologies. Existing XML-based tools required a thorough understanding of XML, so most information workers had trouble understanding how to use them. Also, most information workers do not know how to write code and, therefore, could not easily use development tools such as Microsoft Visual Studio. Therefore, it just made sense to create a tool that developers and information workers could use to create forms based on XML and that users could use to fill out those forms. That tool is InfoPath. (In Chapter 1, we'll tell you exactly what InfoPath



is all about and introduce you to the extensive feature set included in this application.)

Looking at the wealth of features included in InfoPath, especially those added in InfoPath 2010, it also made sense to create this book. This book is titled *Designing Forms for InfoPath and SharePoint* for a reason. It's all about designing forms using InfoPath Designer 2010, as we're sure you have figured out by now. This book will teach you everything you need to know about creating forms using InfoPath Designer 2010 and probably a few things you never thought you needed to know.

Who Should Read This Book

Whether you are an information worker who has created only a few forms in Word or a software developer who is familiar with more advanced coding concepts, if your intention is to learn how to design InfoPath forms, this book is for you. This book will talk about not only the basics of designing forms but also such advanced concepts as writing managed code for InfoPath. As long as you have an understanding of basic form concepts and a desire to learn, you are in the right place. If you want to learn everything you can about InfoPath 2010, you have found the right book.

How This Book Is Organized

This book contains two parts. Part I is all about designing forms in InfoPath Designer. It introduces the concepts and explains most of the principles needed to design rich forms. No prior coding experience is required to understand the concepts, so both information workers and developers can use Part I to learn the basics of InfoPath form design. Many chapters build on previous chapters and become slightly more advanced as you progress. For example, in Chapter 5 we discuss advanced controls and customization, but by Chapter 7 we show how to pull external data, such as from a Web service, into your forms. By the time you finish reading Part I, you should know everything you need to know to design an InfoPath form for the InfoPath Filler application or for SharePoint Server without having to write any code.

Part II is about advanced form design. In this part of the book, we talk about using more advanced form design techniques, including how to write code for InfoPath. These chapters are geared mainly toward software developers who have some basic coding experience. However, if you are an information worker and you have completed Part I of this book, the second part may interest you as well. In Part II, we talk about such topics as the InfoPath object model (Chapter 13), advanced topics regarding InfoPath Forms Services (Chapter 14), and ways to host InfoPath (Chapter 15).

Conventions Used in This Book

We use a few typographical conventions throughout this book. **Bold** text indicates key topics or terms. The names of features shown in the user interface, such as menu items, appear in *italic* text.

Information that pertains to InfoPath Forms Services is clearly displayed as features in the text. These tips will let you know when certain InfoPath features work differently in browser forms or don't work at all.

Samples

Almost every chapter in this book has one or more samples, which you can download from the Addison-Wesley Web site for this book. Sometimes the samples are InfoPath form templates (.xsn) files, which is the case throughout Part I. In order to use these form templates, you first need to open them in InfoPath Designer and resave them to a local folder. (This will make more sense after you start reading Part I.) Trying to open the form template in order to fill it out without first saving it will result in an error.

Some samples include form (.xml) files in addition to form templates. To open the forms, first open InfoPath, and then open the XML file using the standard *Open* dialog. The first time you open one of the sample forms, the dialog shown in Figure P.1 will be displayed. This dialog allows you to choose the form template associated with the form you are trying to open. The text of the chapter indicates the correct form template to use. After you choose the form template, click the checkbox *Always use this form template*

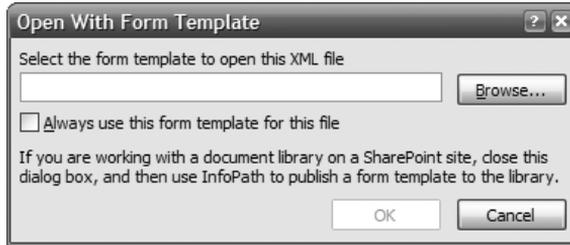


FIGURE P.1: Open with Form Template Dialog

for this file. After the form is opened, immediately save it. This will prevent you from having to choose the form template each time you open the form.

Some sample form templates define one or more data connections. For these samples to work properly, the external data source must exist. To see if a form template depends on a data connection, go to the *Data Connections* menu item under the *Tools* menu while in design mode. Since there are many types of data connections, we'll describe how to set up each one to successfully preview the form.

- XML document: If the XML (.xml) file exists within the form template (under the *Resource Files* menu item on the *Tools* menu), there is nothing you need to do. If the XML file is external to the form template, you will need to click the *Modify* button on the *Data Connections* dialog to point to your copy of the XML file. You can find the file within the samples for a given chapter.
- Database: A database connection depends on a SQL Server or Access database. If the sample uses a SQL Server database, you must have SQL Server installed and have administrative rights to the SQL Server instance. For an Access database, the chapter will include an Access database (.mdb) file. For either case, click the *Modify* button on the *Data Connections* dialog to update the data connection to point to your database to restore the connection.
- Web service: To use a Web service, you must have Internet Information Services (IIS) 6.0 or later installed on your computer. The ASP.NET 2.0 ISAPI Web extension must be enabled, and ASP.NET should be configured to render .aspx pages. Copy the Web service code from

the sample and paste it into a new ASP.NET Web service project in Visual Studio 2005. If you don't have Visual Studio, you can still create the Web service by creating a text file with extension `.aspx` within an IIS virtual directory. Check to see if the Web service code requires read or write access to specific directories; you can grant access to those directories or simply update the code to use directories of your choice. Before using the Web service with InfoPath, try navigating to the Web service by using a Web browser on the local machine. Once the Web service works outside of InfoPath, you can click the *Modify* button in the *Data Connections* dialog to change the Web service connection from the sample to point to your own Web service.

- SharePoint library or list: The prerequisite to using samples with a SharePoint library or list connection is a Microsoft Office SharePoint server. If you have a SharePoint server, ensure you have at least reader rights if the connection is only reading data. Likewise, if the sample form template submits to a SharePoint library, you must have contributor or higher privilege. Ensure that the library or list upon which the form template depends actually exists on the server. To use your library or list, edit the connection to point to your server and select the appropriate library or list.

In Part II, most of the samples include code. Those samples that require you to perform special actions (or actions in a specific order) to build the samples include `ReadMe.txt` files that explain what you need to do.

For sample form templates that include form code, you must do the following. First, find the sample form template (`.xsn`) and archive (`.zip`) files with the same name. Extract the archive to a location on your computer by right-clicking on the `.zip` file and selecting *Extract All*. Next, open the form template in design mode as you would for samples without form code. Select the *Microsoft Visual Studio Tools for Applications* (VSTA) menu item under *Tools* and then *Programming*. If InfoPath cannot find the VSTA project, the dialog shown in Figure P.2 appears. Click the *Browse* button to navigate to the *Visual C# Project* (`.csproj`) file within the extracted folder. Once the *Microsoft Visual Studio Tools for Applications* window appears, hit F5 to

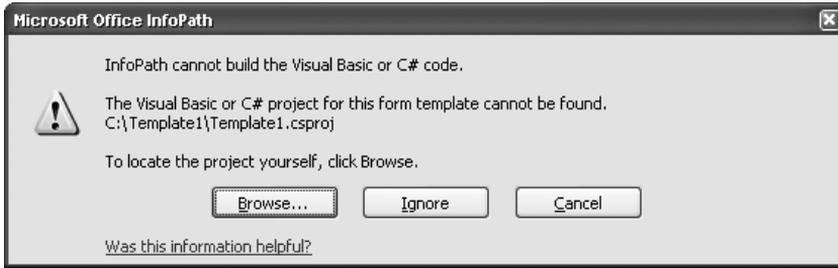


FIGURE P.2: Dialog shown when InfoPath cannot find the Visual C# project with the form code

fill out the form while previewing it. The debugger will be automatically attached, so any breakpoints or unhandled exceptions halt form execution.

Some samples in Part II require references to the interop assemblies for InfoPath. In order to set a reference to the correct interop assemblies, open the *Add Reference* dialog in Visual Studio and browse to the install location for Microsoft Office 2010. (This is usually C:\Program Files\Microsoft Office\Office14.) Then, locate the *ipeditor.dll* file and select it. This will include the *Microsoft.Office.Interop.InfoPath* interop assembly that you need as well as a few assemblies that aren't needed. In order to be able to install the samples, you will need to remove the references to the *ADODB*, *MSHTML*, and *MSXML2* assemblies. Some samples use the *Microsoft.Office.Interop.InfoPath.Xml* assembly. You can also locate this assembly in the install location for Office 2010.

Some code snippets that you see within chapters may differ from the code in the sample. Due to space constraints, brevity in code may have resulted in reformatting or removal of comments or error-handling code that are not required to understand the sample. Regardless, the functionality of the code itself remains unaffected. Note that the code included with the sample form templates is not considered production quality. The code samples, not to mention the form templates themselves, have not been subjected to the rigorous testing you would expect from a company such as Microsoft.

This page intentionally left blank



Acknowledgments

The process of writing a book is never a one-person job (or three-person job, in this case). There are always many different people involved in the writing of a book, from the publisher to the reviewers to the product team whose product we are writing about. Therefore, we extend heartfelt thanks to the following people.

First, we want to thank the entire InfoPath product team. Without the InfoPath team and the wonderful application that is InfoPath, this book wouldn't exist. Next, we thank our editor, Joan Murray, who has guided us throughout the 15 months it took to make this project a reality. Also, the following people from Addison-Wesley played a key role in the process: Curt Johnson and Eric Garulay. We also thank our copy editor, Kelli Brooks, who did an outstanding job, and our project editor, Anna Popick.

Hagen would like to thank Scott for his mentorship throughout writing this book. He is not only a role model but also a great personal friend. I could not have asked for a better coauthor as well as coworker. Scott, thank you for all you have taught me in authorship.

Scott would like to thank Hagen for his dedication and persistence throughout the long and difficult process of completing this project. Your hard-working attitude and easygoing nature make you a joy to work with. We started this project as close personal friends and made it through all the tough times unscathed.

Jessica Meats would like to thank Nick Dallett and his team, as well as others at Microsoft for their comments and feedback throughout the writing process. Thanks goes out to Christopher Brotsos, Bojana Duke, Laura Harrington, Anson Hidajat, Silviu Ifrim, Petru Moldovanu, Philip Newman, Shirish Pulikkal, Roberto Taboada, Andrew Whitechapel, Daniel Witriol, and Richard Witte. She would also like to thank Chris Marchal for his help around the new authentication options in SharePoint 2010. Also to Michael Lotter for his contribution of content around the SharePoint and InfoPath integration content that's included in Appendix A of the book.

Last, but certainly not least, we'd like to thank our families. Scott thanks his wife, Andrea, and his two sons, Sean and Bradley, for being so supportive during the 15 months it took to write this book. Without the three of them, life would have been much more difficult during this time. Hagen thanks his girlfriend, Jaime, for being by his side throughout one of the biggest undertakings of his life. Hagen also thanks his parents, Christine and Stuart, for their constant encouragement and support, especially his dad for being his biggest fan and the first to preorder this book!

Feedback

After you read this book, we would love to hear what you think about it—both positive and negative. This will help us improve future revisions. Also, feel free to send us any questions you may have. We would be happy to help. To contact us, simply send us an e-mail at DesigningIPForms@hotmail.com. If you have a question, we will reply as quickly as possible. We hope that you enjoy the book and that it makes the process of designing InfoPath form templates much easier.



About the Authors

Scott Roberts is a Principal Development Lead in the SQL Server group at Microsoft Corporation. Previously, he was a Development Lead on the InfoPath team, and was involved with InfoPath since its inception. While in the InfoPath team, he led the development of many features for designing and editing InfoPath forms. Scott is also the author of *Programming Microsoft Internet Explorer 5* (Microsoft Press, 1999) and numerous technical articles and publications.

Hagen Green is a Senior Program Manager at Microsoft Corporation, working on the Windows Phone Communications team. He previously led the User Experience Platform test team in SharePoint Foundation. He contributed chapters on InfoPath to *Visual Studio Tools for Office: Using Visual Basic 2005 with Excel, Word, Outlook, and InfoPath* (Addison-Wesley, 2006), and *Visual Studio Tools for Office: Using C# with Excel, Word, Outlook, and InfoPath* (Addison-Wesley, 2005).

Jessica Meats is a Partner Technology Advisor for Microsoft Corporation specializing in Microsoft SharePoint Server, Office and Business Intelligence. She joined Microsoft in 2008, straight out of university, as a Partner Technology Specialist and chose InfoPath as one of her specialities, wishing that some of the companies she'd done temp work for as a student



had used such a product. Her job involves helping partner organizations work with these products and aiding new partners quickly gain the skills needed to build a practice around delivering these solutions. Jessica is the author of a science-fiction thriller, *Child of the Hive* (Book Guild Publishing Ltd., 2009). In her spare time, Jessica writes fiction, juggles fire, and studies kung fu.

This page intentionally left blank



1



Introduction to InfoPath 2010

What Is InfoPath?

The rapid adoption of XML-based technologies over the past decade or so has precipitated the need for a tool that helps end users interact with and share XML data. On August 19, 2003, as part of the Microsoft Office System 2003, Microsoft shipped a new application created to fill that need—Microsoft Office InfoPath.

InfoPath may be one of the least known yet most appropriate platforms for gathering data in the Microsoft Office suite of applications. But InfoPath's popularity has been on a sharp rise. Anyone from an information technology (IT) manager tracking a purchase request workflow to little Johnny tracking his music collection can benefit from what InfoPath offers. Before InfoPath, programs such as Word, Excel, or Access may have been used for these tasks. There is nothing wrong with using any of these other programs to build forms. However, none of these applications were built with forms in mind, so they don't provide the ease of use and power that InfoPath has when it comes to creating a form based on XML data. The goal of this book is to make you an expert InfoPath form designer. Whether you are a novice, an information worker, an experienced developer, or anyone in between, you have found the right book to help you get the job done with InfoPath 2010.

You may be wondering why Microsoft created this new tool. Doesn't Microsoft already have other form design tools such as Word, Excel, and Visual Basic .NET, to name a few? While it is true that you can build forms with these tools, none of them are based solely on XML technologies, nor were they created exclusively to edit XML data. Both Word and Excel can interact with XML data, but they do not offer structural editing of the XML data to the extent that InfoPath does. Visual Basic .NET requires you to have at least a basic understanding of application development. InfoPath does not require you to be a developer in order to build a form. Anyone who has experience using Word can create a form in InfoPath Designer.

Thanks to the familiar Microsoft Office user interface (UI), ramp-up time to productivity is minimal. Users with little or no programming experience can use InfoPath Designer to create form templates. InfoPath Filler users can also fill out forms based on those templates. Since InfoPath was designed (no pun intended) to be used by almost anyone with little or no training, form designers have all the tools needed to deliver form templates that are easy to use. Of course, if you are a developer, you may also want to customize your form templates, for example, by writing script or managed code using InfoPath's extensive object model (OM).

Form Templates and Forms

Form template: The blueprint for a form, which you create in InfoPath Designer. Many forms that are unrelated in content can be created with the same form template. The form template defines the look-and-feel and functionality of a form.

Form: An instance of a form template that users fill out by using InfoPath Filler or an Internet browser connected to InfoPath Forms Services.

Using InfoPath Designer, you can rapidly build form templates that interact with XML data and XML Schema (XSD) with little or no code required. Then users can open InfoPath Filler to fill out forms based on the form templates you created. Because InfoPath is a Microsoft Office

product, the UI is as familiar to users as Word. In fact, InfoPath provides a lot of the same features as Word (e.g., spell checking, rich text editing), so filling out forms in InfoPath is as easy as drafting a document in Word. The main difference is that when users are filling out an InfoPath form, they are actually editing XML data.

However, the value of InfoPath goes way beyond its ability to edit XML data directly. In fact, its usefulness as a data-gathering and management tool is one of the most powerful aspects of InfoPath. Also, because the InfoPath form templates you create can be based on industry-standard XML Schemas and integrated with Web services, data stored in InfoPath forms can be integrated with existing processes in your organization and across other organizations.

Let's look at an example. Say that you run a sales organization. You have an existing database that contains customer and sales data. Your sales manager wants you to create two different forms—one for salespeople to input customer information and another for them to input sales data. Before InfoPath, there were many ways to create these forms. However, since the data comes from a back-end database, all of these methods involved writing some code. And you would typically have to duplicate this code from one form to the next.

With InfoPath, creating these forms is much easier. InfoPath allows you to create form templates that connect to existing databases or Web services, thus removing the need to write code to access the data. This part is done for you. InfoPath also allows you to easily merge the data gathered in these forms so that you can create aggregate reports for upper management. Again, InfoPath enables you to do this without writing any code at all, unlike other form tools. InfoPath includes an extensive OM, so it's highly customizable. Using InfoPath, you can enable almost any data collection scenario you can dream up. The scenarios can be extended by combining InfoPath with SharePoint workflows to rapidly develop applications to automate business processes in the enterprise. Deeper integration with SharePoint is a key development point in the latest release, which makes it easier to link your data collection forms to the processes they drive.

As your business needs change, so should your software. InfoPath is designed to cater to these ever-changing data collection needs. Built on

World Wide Web Consortium (W3C) standards, XML and its associated technologies help InfoPath deliver on its promises. A few of the greater benefits of XML include the following:

- Provides a semistructured data format
- Is human readable
- Is easy for programs to process
- Works across disparate systems

Technologies encompassing XML include Extensible Stylesheet Language Transformations (XSLT) for transforming the XML to another format such as HTML, XML Path Language (XPath) for selecting specific parts of the XML data for processing, and XSD for defining precise rules for how the XML structure and data should look. InfoPath seamlessly incorporates each of these technologies into the experience of designing and filling out forms. For example, thanks to XSLT, one of the most important benefits of using InfoPath is the ability to separate data from how it is displayed. With the drag-and-drop UI of the InfoPath Designer, you can completely change the look-and-feel of a form without changing the underlying data that is saved when a user fills it out. Another advantage is being able to merge data from multiple forms. This is especially useful when creating a report, for example.

As we mentioned, InfoPath offers a plethora of built-in features geared toward collecting data that would require custom code in other form design tools. With InfoPath, for example, you can use data validation and conditional formatting to ensure that users enter valid data before the form is submitted to your back-end databases or Web services. You can also design your forms to collect only the data you need. Traditional forms include many parts that don't apply to everyone. For example, employment applications often contain sections relevant only to people with college degrees. However, these forms have to include sections for this information since the forms are static by nature. These forms may contain instructional text such as "If you chose Yes for question 5, please enter more information here. Otherwise, skip to question 6." With InfoPath, you can design dynamic forms that contain the additional sections only if needed.

Using our example, when a user chooses Yes to question 5, a section of the form that wasn't visible suddenly becomes available. Also, the static employment application may not provide enough room to fill in a complete employment history. In this case, users must either not include the additional information or provide it on an additional form. With InfoPath forms, your users can fill in as much data as they need since your forms can be built to grow dynamically. When your users need more space to fill in additional data, the form can expand to meet their requirements.

These are just a few aspects of InfoPath that makes it a more powerful tool for XML-based forms than other applications. We will talk about these benefits and more throughout this book. First, however, we'd like to give you an overview of the evolution of InfoPath through four successive versions—InfoPath 2003, InfoPath 2003 Service Pack 1, InfoPath 2007, and InfoPath 2010. This will not only give you some insight into the product's background but also introduce you to several of its features.

InfoPath 2003

To provide reasonable context for discussing new InfoPath features, let's look at the different versions of the product and what each brings to the table. Version 1, officially released as InfoPath 2003 on August 19, 2003, is Microsoft's inception into the XML forms and data collection markets. As such, it bundles competitive features to allow for a rich form experience. This section recaps the highlights of InfoPath 2003.

The InfoPath design mode is all about creating nice-looking forms that include basic functionality. The design surface, or view, is similar to that of Microsoft Word. The content of the view flows from left to right, top to bottom, and serves as the presentation layer for the form's data. Absolute positioning, such as what PowerPoint offers, would make InfoPath form design cumbersome when accommodating different resolutions and monitor sizes. Therefore, **layout tables** are used instead as the positioning paradigm of choice—equivalent to that of many professional Web sites. Tables offer a familiar and flexible option when demanding pixel-by-pixel, picture-perfect parity.

For those of us with a less than acute sense for visual design, **color schemes** spice up even the most boring forms. A single click on the *Bright Blue* color scheme adds a subtle blue color theme to parts of your form template's controls and layout tables. If you don't care for the effects of *Bright Blue*, you can choose another scheme at your convenience at any time. Color schemes more dramatically affect Repeating Table controls more than other controls. The header and footer backgrounds, text, and table borders comply with the current theme.

InfoPath 2003 introduced a general set of controls for inclusion in forms. Figure 1.1 shows the *Controls* task pane. As you would expect, the InfoPath design mode includes controls such as Text Box, List Box, Drop-Down List Box, Check Box, Bulleted List, Numbered List, Plain List, Option Button, and Button. The Section, Optional Section, Repeating Section, and Repeating Table controls support InfoPath's structural editing features. These

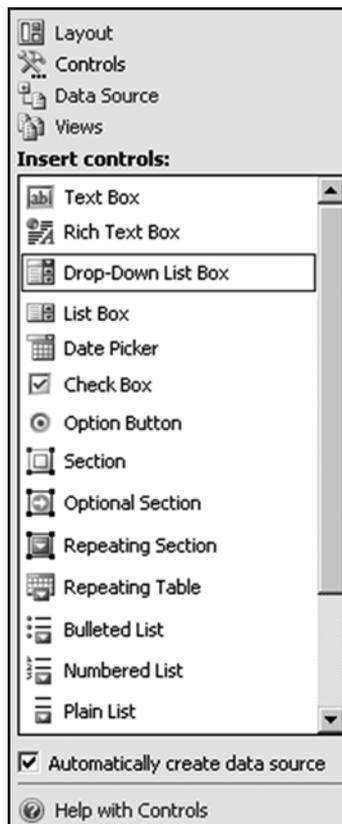


FIGURE 1.1: Controls task pane in InfoPath 2003

editing features allow the end user, while filling out the form, to modify the structure of the XML data as the XML Schema allows. Structural editing is what makes InfoPath such an attractive offering compared with its alternatives. The Hyperlink, Expression Box, Picture, and Rich Text Box controls round out the controls available in InfoPath 2003.

We mentioned earlier that the data and the presentation layers are two disparate concepts. We've already discussed the presentation layer, but we haven't yet focused on the data. When we refer to "data," we really mean "data and structure." XML consists of hierarchical elements that may or may not contain text content. The definition for the XML structure is bounded by the rules of the XML Schema. (We will talk more about XML Schema in Chapter 4.) InfoPath abstracts away the XML Schema details through the *Data Source* task pane (DSP; see Figure 1.2). The DSP is a design-mode feature that shows what the XML Schema allows. If you don't have

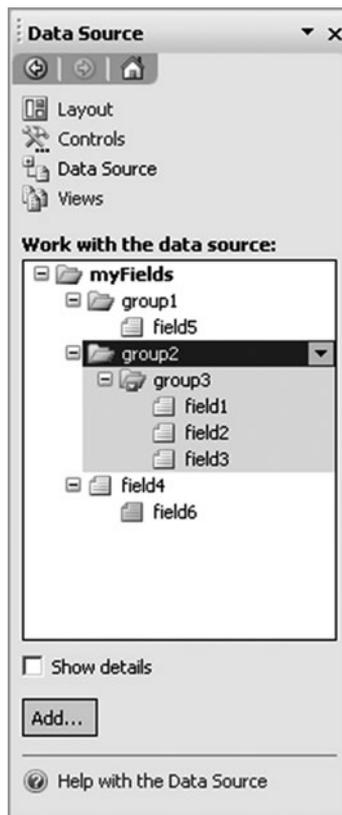


FIGURE 1.2: Data Source task pane in InfoPath 2003

an XML Schema lying around, there's no need to worry. InfoPath automatically builds the XML Schema as you insert new controls onto the designer view. Inserting a control into the form may create either a **field** or a **group** in the DSP. This process populates the DSP based on which controls are selected. The newly inserted controls are **bound** to the newly created data source. The word **binding** describes the connection between what the user sees and what data is stored behind the scenes. We'll talk more about binding in Chapter 4.

Binding sounds easy, right? It is—for most situations. But due to the flexibility of hooking up any data source to any controls, getting into odd binding dilemmas is inevitable. **Design-time visuals** are icons displayed on your form controls as you design your form template. These visuals serve as handy reminders to a designer that the form may be broken. We will discuss this versatile feature in more detail in Chapter 4.

Besides the visual zest, let's look at the different ways to pull external data into an InfoPath 2003 form. **Secondary data sources** are used to connect to databases, SharePoint libraries or lists, and Web services. Say your car registration form template has a Drop-Down List Box control containing all 50 states in the United States. When creating your form template, you could enter each state manually when creating the list items for the control, but it's probably easier to point the Drop-Down List Box control to an existing database. SharePoint servers are somewhat ubiquitous in high-tech small business. Pulling a real-time list of manager names or part numbers into your form allows it to always be up to date with no intervention on your behalf.

Web services have created a growing sensation throughout the Internet community for the last several years. Any popular search engine will yield innumerable results when you query for Web services. Ironically, many search engines also expose a Web service for you to do the searching. InfoPath supports connecting to these Web services without you, as the form designer, writing a single line of code. All you need to know is where the Web service resides. So forget about details like the Web Service Definition Language (WSDL), input parameters, and output parameters that you would need to understand if using other form tools. However, if you wish to deal with those aspects of the Web service using InfoPath 2003, you can access them by writing script code against the InfoPath OM.

The InfoPath OM in version 2003 is designed as a programmatic extension to many features provided in the InfoPath UI and some features that are not. The OM is not a superset but rather a supplement to the functionality available from the UI. To access the OM in InfoPath 2003, you could choose to write either JScript or VBScript in the Microsoft Script Editor (MSE). MSE is the Microsoft Office–bundled development environment of choice for scripting languages. InfoPath even has an extensive help system built into MSE that provides on-demand, context-sensitive help while scripting InfoPath objects.

So let's say that we've made the final touches on our new InfoPath form template by adding some JScript code using MSE. We're all ready to roll the form template out to our team. It's time to **publish** our form template. Publishing is a short process to deploy the form template to a location accessible by users. A few steps through a wizard is all it takes to properly put your template on a network file share, SharePoint site, Web server, or other shared location.

We've exhausted our brief talk of the InfoPath design mode for now. Let's peek at some of the important features you can find while filling out a form. As mentioned earlier, the UI is instantly familiar. You can expect the standard Microsoft Office set of features, such as cut, copy, and paste; a dizzying array of font formatting options; spell checking; find and replace; and clip art (to name a few).

What's so special about InfoPath? It sounds a lot like Word in many ways. However, InfoPath has two key advantages: An InfoPath form uses XML data, and InfoPath offers data validation, which is not found in Word. If you are an application developer, you will appreciate the tight integration between XML data type validation and the InfoPath UI. You can think of data validation as a way to guide your users to format their input as you determine. For example, if you want a phone number, you could easily use a text box that accepts any text the user types. But with data validation, you can disallow a phone number if it's not ten digits or has non-numeric characters. InfoPath's validation model is also extensible by using the InfoPath OM and writing custom form code.

A popular way to make your form template more interactive is to use **conditional formatting**. Take an expense report form that adds up prices

from a list of items purchased during a business trip. You can design the form template so that if the sum of expenses is more than \$1,000, the text for the total value will change from a calm black to an attention-getting bold red. Conditional formatting can be used throughout your form template to not only change fonts but to show and hide regions of your form as well.

It's great to have your users happily filling out your forms. But how do you make sense of the data saved in XML files? InfoPath has convenient, built-in reporting features to bring together data for you and your users. The first of these reporting functions is form merge. When a user selects multiple saved forms, InfoPath aggregates the data from each saved form into a single new one. A common use for this feature is rolling up status reports for the boss. InfoPath also exposes custom form merging for advanced developers by allowing them to write their own custom XSLT. InfoPath 2007 made some great improvements to this customization experience by supporting custom merge actions in design mode. Another way to report InfoPath form data is to export it to Excel. Sometimes InfoPath may not be the best environment to do complex spreadsheet functions like pivot tables. Exporting the data to Excel is a simple click away, and you can customize the process by exporting specific parts of the data. Once data is exported to Excel, you can create pivot tables and charts to help you analyze the data. Since InfoPath is a powerful data-gathering tool, reporting options such as form merging and export to Excel help round out the story for bringing data together in a useful way.

InfoPath 2003 Service Pack 1

Even before InfoPath 2003 was out the door, InfoPath 2003 Service Pack 1 (SP1) was under construction; it was finally released on July 27, 2004. Convenience features, such as a table-drawing tool and the Format Painter, were added in SP1. Integration was introduced or enhanced, such as Web service support for ADO.NET DataSet objects. New controls, such as the File Attachment and Master/Detail controls, were added. Finally, InfoPath was also made more robust by adding features like crash recovery while filling out a form. Even if the computer doesn't crash, there's always the

possibility that a power outage may occur (or the battery may die on your laptop). If you didn't think it could get any better, InfoPath 2003 SP1 was available as a free upgrade on the Web and fully backward compatible with 2003.

In tune with the numerous enhancements you would expect from a new release, InfoPath 2003 SP1 provided a much richer design experience. A lot of features that were already available in other Microsoft Office applications were integrated into InfoPath 2003 SP1. The new *Insert Layout Table* button (available on the *Standard* toolbar, as shown in Figure 1.3) enables you to quickly insert a table with as many as five columns and four rows.

Another tool that was added in SP1 to make table creation easier is the new *Tables* toolbar (see Figure 1.4). This toolbar, which includes a table-drawing tool that can be turned on by clicking the *Draw Table* button, simplifies the process of creating tables tremendously. Using the table-drawing tool, you can easily draw a table any way you want without having to use the *Insert Table* dialog. The *Tables* toolbar also allows you to quickly and easily change various properties of any table, such as borders and shading. Considering that most InfoPath forms rely heavily on table layout, these tool improvements were welcome additions.

Another very useful tool added in SP1 is the **Format Painter**. You may already be familiar with this tool since it's also available in other Microsoft



FIGURE 1.3: Insert Layout Table toolbar item



FIGURE 1.4: Tables toolbar

Office applications, such as Word. This tool allows you to quickly copy formatting from one part of the form template's view to another. You can select text or a control and copy its formatting to other text or another control in the form template.

The new table tools and the Format Painter are just two of the many features added to InfoPath 2003 SP1 in response to customer feedback. There are many more features that were added as well. Another feature added in response to feedback from international markets is support for complex script and right-to-left (RTL) languages. With InfoPath SP1, you can create forms that work with languages requiring complex script and/or RTL reading such as Arabic, Hebrew, Chinese, and so on.

InfoPath 2003 SP1 also added a slew of new controls to the *Controls* task pane. As with the other features added to improve the design experience, the decision about which controls to add in SP1 was based solely on customer feedback. (We will talk about each of these controls in more detail in Chapters 2 and 4.) The following controls were added in InfoPath 2003 SP1:

- Master/Detail
- File Attachment
- Choice Group
- Repeating Choice Group
- Choice Section
- Recursive Section controls
- Scrolling Region
- Vertical Label
- Ink Picture
- Custom controls

One of the most exciting additions to the *Controls* task pane was the support for custom controls. As of InfoPath 2003 SP1, you can use ActiveX technologies to build your own custom controls for InfoPath. Support for custom controls expanded in InfoPath 2007 and further expanded 2010 to include the ability to build controls without code.

Data source enhancements, such as support for choice and recursion, were added in conjunction with the new Choice Group and Recursive Section controls. Customers clearly use industry-standard XML Schemas with choice constructs. As a result, InfoPath 2003 SP1 fully supports the XML Schema choice element and the Choice control that binds to it. Recursion is also very common in real-world XML Schemas. Recursion is when a schema element can contain itself. For example, a schema describing a hierarchy of employees can be described only by using recursion (since an employee can have employees under him or her). The Recursive Section control naturally supports recursion by allowing you to insert the same item as a child of itself. In SP1 there is no extra burden on the form designer to get this working properly, but in InfoPath 2003 it was a bit of a technical hurdle.

One of the most useful features introduced for nondevelopers in SP1 was the **Rules** feature. Overwhelming customer feedback was clear: Too many seemingly simple things required writing script. For example, showing a dialog or changing views required a few lines of script. With rules, setting up your form template to show a dialog is a few button clicks away. A more complex scenario would be to switch views, submit to a database, and set the value of a field—all in one rule and without code! Introducing rules to the masses meant less programming and more form function.

Creating a rich, functional form template is sufficient until you realize that your one form template does not fit all users. Sometimes data in the form shouldn't be shown to specific users. Imagine a form that tracks top issues, and each issue has a checkbox to set the entry as private. With InfoPath 2003 SP1, you can assign users to private and public **roles** and, in combination with conditional formatting, decide whether or not to show private data based on the current user role.

When it comes to trusting the data in a form, little can supersede the security offered by **digital signatures**. Highlights such as nonrepudiable partial signatures, cosigning, and countersigning allow the form template designer a greater level of freedom while not sacrificing the concept of "secure by default." Factor in the extended programmability support for digitally signed forms, and you have a complete solution for protecting form data.

Arguably one of the most anticipated features in the SP1 release was the introduction of **managed code** support. With the world buzzing about .NET, customers could not wait to get their hands on the free Visual Studio Toolkit plug-in. Along with managed code support, a revised and expanded programming interface, or object model was included. In Chapter 13 we will introduce the OM, which was introduced in InfoPath 2007.

Not only can InfoPath simplify the creation of forms but, as mentioned earlier, since InfoPath forms are based on XML, they can be integrated easily into existing business processes and workflows. By using InfoPath with Microsoft Office SharePoint Server or Outlook, you can quickly build workflow processes in a fraction of the time it would take with other forms development tools. We will talk in more detail about workflow in Chapter 10.

With the growing popularity of Tablet PCs, it seems only natural that InfoPath would include support for these devices. InfoPath 2003 included limited support for Tablet PCs that was expanded in SP1. In design mode, the Ink Picture control enables you to include sections in your form template that are exclusively for handwriting with the Tablet PC pen. When editing the form, all controls support an ink-entry mode that allows you to write in them with your Tablet PC pen. Your writing will then be converted to text. While in ink-entry mode, InfoPath supports many of the same gestures to edit and correct text you may already be familiar with.

SP1 also brought enhancements in form template deployment. InfoPath 2003 made it very difficult to send form templates by e-mail for users to fill out. Form template designers had to jump through a few technical hoops to get it to work properly. It was also inconvenient for recipients as well. The addition of **e-mail publishing** in SP1 opened up the potential for anyone with an e-mail address and InfoPath to fill out a form. Deploying a form template via e-mail is only a quick click away. Once a template is deployed via e-mail, users can open the e-mail message in Outlook and start filling out the form.

In addition to enabling you to build more elaborate form templates, new features also enhance the process of filling out forms. One very useful new feature is the *Fill Out a Form* dialog. This dialog, shown in Figure 1.5, gives you a starting point for editing forms and a central location for managing them. (These capabilities have now been included in the backstage

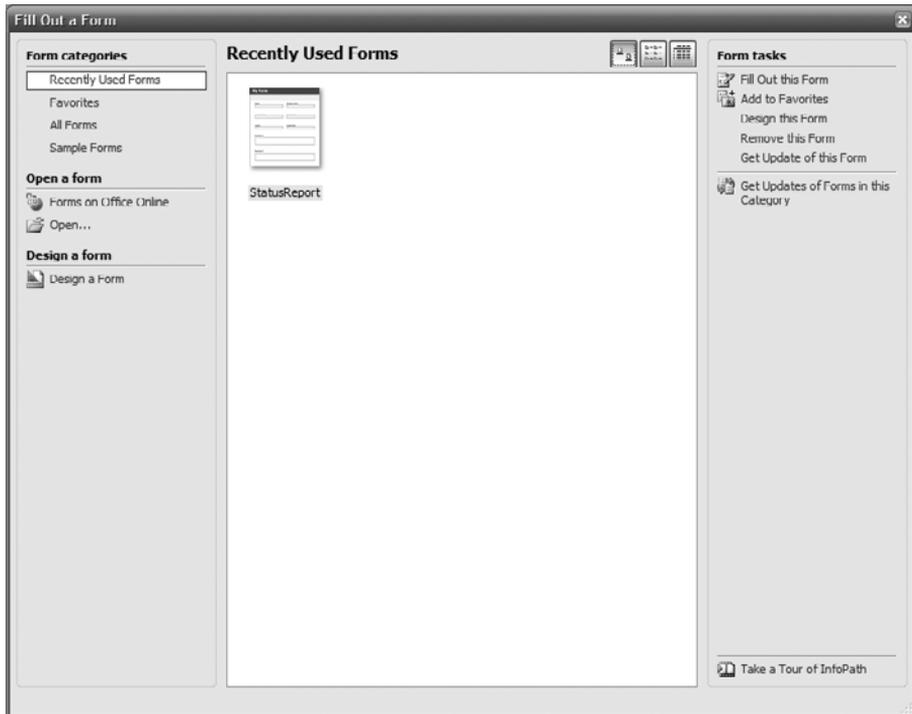


FIGURE 1.5: Fill Out a Form dialog in InfoPath 2003 SP1

view available from the *File* tab.) Using the dashboard, you have easy access to forms you've filled out before by clicking the *Recently Used Forms* link in the left-hand side of the dialog. You can also add forms to your list of favorites by clicking the *Add to Favorites* link on the right-hand side and keep track of your favorites via the *Favorites* link on the left. You can download forms from Office Online by clicking the *Forms on Office Online* link or design a new or existing form template by clicking either the *Design a Form* or *Design this Form* links. This new dialog greatly simplifies the process of creating, filling out, and managing forms.

InfoPath 2007

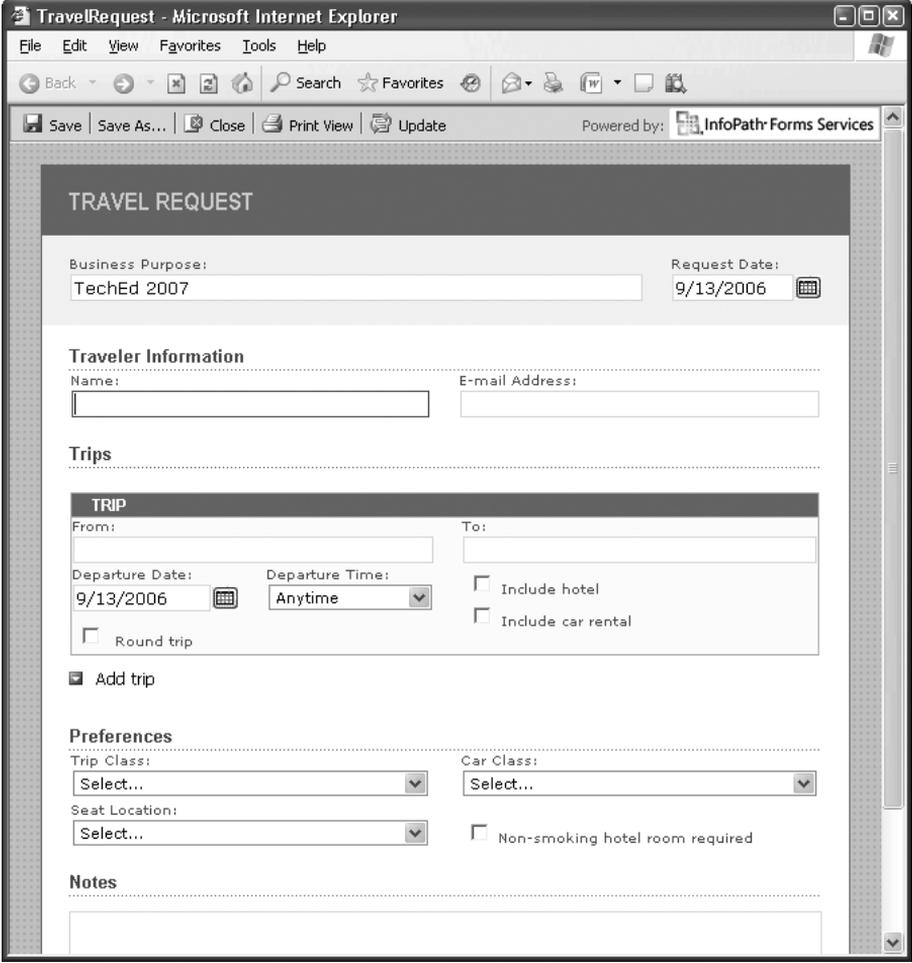
The next version of InfoPath was InfoPath 2007, released on November 30th, 2006. As with previous versions, all feature additions in InfoPath

2007 were based on customer feedback. InfoPath 2007 not only included improvements in the core client application but also finally satisfied the number one customer request—the ability to fill out forms without having to install the InfoPath client application on every user’s computer. To satisfy this request, InfoPath 2007 introduced a sister product called InfoPath Forms Services, included as a feature of Microsoft Office SharePoint Server 2007 Enterprise.

InfoPath Forms Services brought reality to the ultimate deployment story: publishing your InfoPath form template to the Web. Anyone anywhere, on any device, could now fill out your form. Feedback dictated that the InfoPath Forms Services would need to reach UNIX users running Mozilla, Mac OS folks with Safari, and even always-on-the-go businesspeople with their HTML-enabled PDAs and smartphones. Forms were filled out in full fidelity with data validation, conditional formatting, and even custom form code, to name a few features. (Figure 1.6 shows an InfoPath form being filled out in a browser.)

The key to making InfoPath Forms Services successful was to maintain as much compatibility as possible with the InfoPath client application. As a result, most form templates created with the InfoPath client can simply be reused “as is” in Forms Services. The InfoPath 2007 client application supports the few limitations of InfoPath Forms Services by offering two modes: InfoPath-only or browser-enabled form templates. In the browser-enabled mode, some features that are not supported (either in full or in part) are disabled or removed from the InfoPath design mode UI. When designing a new form template, you choose whether to design a client or browser-enabled form template, with the ability to change it at any time. To support switching from an InfoPath-only to a browser-enabled form template, the design mode offers the **Design Checker**, which reports form features added during InfoPath-only mode that are incompatible with InfoPath Forms Services. The Design Checker also helps maintain compatibility with previous versions of InfoPath.

You can expect your browser-enabled forms to behave similarly in InfoPath Forms Services as you would with the InfoPath client application. All conditional formatting, data validation, rules, multiple views, digital signatures, security settings, and most controls are respected. There are so



The screenshot shows a Microsoft Internet Explorer browser window titled "TravelRequest - Microsoft Internet Explorer". The address bar shows "Powered by: InfoPath Forms Services". The main content area displays a "TRAVEL REQUEST" form. The form has a header "TRAVEL REQUEST" and several sections:

- Business Purpose:** A text box containing "TechEd 2007".
- Request Date:** A date picker showing "9/13/2006".
- Traveler Information:** Two text boxes for "Name:" and "E-mail Address:".
- Trips:** A section with a "TRIP" header and several fields:
 - From:** A text box.
 - To:** A text box.
 - Departure Date:** A date picker showing "9/13/2006".
 - Departure Time:** A dropdown menu showing "Anytime".
 - Round trip
 - Include hotel
 - Include car rental
- Add trip
- Preferences:** Three dropdown menus for "Trip Class:", "Seat Location:", and "Car Class:", each showing "Select...". A checkbox for "Non-smoking hotel room required" is also present.
- Notes:** A large empty text area.

FIGURE 1.6: Filling out a browser-enabled form template in Internet Explorer

many similarities between the InfoPath Forms Services and the InfoPath client application that it's a more interesting conversation to discuss what the InfoPath Forms Services in the 2007 version *does not* support.

Due to the InfoPath form architecture and the tremendous flexibility in form configuration, some of the more advanced features of the InfoPath client are not supported. For example, forms with script are not supported. Managed form code works as expected, but showing UI, such as a message box, by calling methods in the InfoPath or Windows Forms object models, is not. Another major difference between InfoPath and Forms Services is

the behavior of data connections. The security model of data connections is more restrictive with browser forms. Many of the differences between the two types of forms have been addressed in the 2010 release, but the later chapters will cover in more detail which features are still unavailable in the browser.

InfoPath Forms Services is built on top of the rich capabilities of the SharePoint Server 2007 platform, which includes Windows SharePoint Services. You can harness the full power of SharePoint sites, document libraries, lists, and other SharePoint-specific features. The SharePoint Server package adds portal features such as Workflow Services and Enterprise Search. Partially filled forms in the browser can be saved to a form library and ultimately submitted to a SharePoint document library, for example. Administrators can configure InfoPath Forms Services settings from the command line, but the SharePoint Central Administration site offers a more user-friendly option for doing so.

A favorite feature for Web designers is the ability to host a browser-enabled form as a control in a Web page. Since Forms Services is implemented as an ASP.NET control, Web design gurus familiar with ASP.NET will have no problem incorporating forms into their existing Web infrastructure. Various events exposed from the form control serve as liaisons between the control and the hosting page. This helps tie your form intimately into the overall design and experience of your Web page.

The InfoPath Forms Services package is meant not as a replacement but rather as a complement to filling out forms with the InfoPath client. The relationship between Forms Services and the InfoPath client application can be compared to the relationship between Outlook Web Access (OWA) and the Outlook client application. OWA tries to do the job of the Outlook client on the Web but offers fewer features than the Outlook client application; for example, OWA lacks offline and client-only mail rules. As with OWA and Outlook, when the InfoPath client is available, you would prefer to use it over the server due to the fact that the InfoPath client provides a richer set of features. Even with the advancements in browser and Web technology, it is nearly impossible to do the same things in a Web browser as you would in the InfoPath client. However, most people don't really notice what is missing when they use a browser-enabled form. Limitations

aside, InfoPath Forms Services is a truly remarkable advancement for creating ubiquitous InfoPath forms.

In addition to meeting the number one customer request with Microsoft InfoPath Forms Services, the InfoPath team also responded to feedback about the InfoPath client application by adding many new features.

A *Getting Started* dialog was introduced to give easy access to recently used and created forms. The *Design a Form Template* dialog gives a choice of form types and data sources to simplify the design process. You can customize existing templates or templates downloaded from Office Online. There is an option to import forms from Word or Excel to make use of any existing company forms.

InfoPath 2007 added new controls—the Combo Box, the Horizontal Repeating Table, the Multiple-Selection List Box, and the Horizontal Region. However, the most exciting addition in terms of controls is the ability to create custom controls with no code. InfoPath **template parts** allow you to create custom controls that are aggregates of existing controls already built into the InfoPath design mode. In addition, you can add rules, calculations, data validation, and even secondary data sources to template parts you design, thus creating reusable components with no code required. Another appealing addition to InfoPath 2007 is integration with Outlook. **InfoPath e-mail forms**, as the feature is called, enables users to fill out forms in the body of an e-mail instead of having to open the InfoPath client application.

Some of the other additions to the InfoPath client application are not as obvious. Many performance improvements enhance the usability of InfoPath. Several long operations now offer progress dialogs that can be canceled. Also, cases where multiple dialogs used to be displayed now show only one.

A few more of the many feature additions include Information Rights Management (IRM), date calculations, support for read-only views, bound fields in headers and footers, and better support for offline forms.

Features were also added specifically for developers including integration of the InfoPath design mode into Visual Studio Tools for Office (VSTO). But what if you just want to enhance your forms a bit without installing Visual Studio or resorting to script? You can do that with Visual

Studio Tools for Applications (VSTA). With VSTA, you can use managed code anywhere you would have used script in InfoPath 2003 and SP1.

Another cool feature addition geared toward developers in InfoPath 2007 is support for add-ins. Developers can write their own add-in components using the Component Object Model (COM) or managed code. These add-ins enable you to control or modify InfoPath's behavior and add functionality that does not exist in the core InfoPath client application. Since add-ins are supported by other Microsoft Office applications, developers can write code once that will work across applications. Many developers have asked for the ability to host InfoPath since InfoPath 2003 was released. With InfoPath 2007, it became possible to add rich form editing to your own applications by reusing InfoPath. This is such a popular scenario that other Microsoft Office applications have taken advantage of this feature. Word, Excel, and PowerPoint now show their document properties in a **Document Information Panel** that is actually an InfoPath form. The Document Information Panel includes all the features you have come to expect from an InfoPath form. In the advanced section of the book, we will show you how you can add these features to your own application.

InfoPath 2010

The next wave of Office products, Office 2010, includes InfoPath 2010. Unlike the previous versions, the InfoPath client is split into two programs in this release—InfoPath Designer and InfoPath Filler. This creates a clearer distinction between the process of creating a form and the process of filling one out. There is also a change in focus from InfoPath being a standalone product to being the primary design tool for SharePoint forms. This builds on the browser-based forms capabilities that were introduced in the 2007 release with easier customization of SharePoint data collection elements for lists and workflows.

When opening either product, those familiar with earlier versions of InfoPath will notice instantly the change in user interface. The fluent user interface, sometimes referred to as the ribbon UI, which was introduced in many Office 2007 products, has now extended to InfoPath. The idea of the ribbon UI is to make it easier to find features by grouping them into

contextual tabs that appear when relevant, such as the *Table Layout* tab (see Figure 1.7), appearing when you start to edit a table.

When the ribbon UI was first introduced, a strong piece of customer feedback was regarding the lack of a *File* menu. So, in the 2010 release of Office, a new backstage tab was introduced, which allows access to those functions traditionally associated with the *File* menu, such as *Save* and *Open*, as well as functionality around publishing and compatibility of the form. When InfoPath Designer is first opened, it opens in this backstage view, automatically selecting the section for designing a new form. This replaces the *Design a Form Template* dialog from InfoPath 2007, but many of the template types, such as Blank Form and Web Service, remain the same. The main differences in form template types are around the tighter integration with Microsoft SharePoint Server 2010. There are the options to create forms based on a SharePoint List, a SharePoint Form Library, and a form that sets the document properties of files in a SharePoint document library.

The integration improvements between InfoPath and SharePoint include the ability to customize the SharePoint list entry form using InfoPath and more easily create forms to initialize workflows and execute workflow tasks. One of the improvements around browser-enabled forms is the introduction of an InfoPath Form Web Part to SharePoint, which makes it much easier to include an InfoPath form as part of a SharePoint page. Combining InfoPath with SharePoint will be discussed in more detail in later chapters of this book.

The 2010 release of InfoPath improves the browser-enabled form capabilities provided by InfoPath Forms Services. The gap of functionality has narrowed between what is possible in the InfoPath client and what is possible in the browser. The majority of features and controls now work exactly the same in the browser as the client. When designing a new form, the default setting is for a Web browser form, whereas in InfoPath 2007 the default was for a client form.



FIGURE 1.7: The Table Layout tab

InfoPath 2010 has introduced some new controls—Person/Group Picker, External Item Picker, Picture Button, Hyperlink, and Signature Line.

There are other significant improvements around the management of rules. There is now a *Rules* task pane (see Figure 1.8), which can be used to organize the rules associated with the controls on the form. This makes it easier to add and manage multiple rules—for example, conditional formatting or setting field values—to an individual control. There is also a new option, which is to copy a rule or rules from one control and paste them on another. If you have two text fields and want to create the same multiple conditional formatting rules on both, you can now create the rules once, use the *Copy All* button, and then simply select the other field and select to *Paste Rules*. In previous versions of InfoPath, you would have been forced to create the rules separately for each control.

The developer experience for InfoPath 2010 has had a few changes. As with InfoPath 2007, you can use Visual Studio Tools for Applications



FIGURE 1.8: The Rules task pane prior to rules being created

to add small pieces of code to a form. The ribbon UI includes a *Developer* tab. This allows you to choose from a selection of event templates to easily associate your code to specific actions within the form. For example, selecting *Form Loading Event* allows you to write code when the form is loaded and *Changed Event* creates code associated to a selected field that will run when that field changes. This makes it easier to organize small pieces of code attached to a form.

There are other improvements that are less obvious around performance. Some processes that used to take many dialog boxes now involve less. There is one significant improvement around form publishing, which is the introduction of the *Quick Publish* button. As with earlier versions, when you have finished designing your form, you must publish it before it can be filled out. The various options for publishing will be discussed in Chapter 9. In InfoPath 2007, every time a change is made, the form must be republished by going through the whole publishing process. In InfoPath 2010, the *Quick Publish* button allows you to publish your updated form template while keeping the rest of the settings the same.

As you can probably tell just from this introduction, InfoPath contains a very rich set of features for creating forms for SharePoint Server 2010. In addition, InfoPath remains a powerful tool for creating standalone forms.

What's Next?

You may be thinking, "With the many features available in InfoPath, where do I start?" That's a good question. The answer obviously depends on your level of expertise. If you have no experience with InfoPath, start at the beginning. Part I is all about designing form templates from scratch and assumes you have no prior knowledge of InfoPath. Starting with Chapter 2, we will walk you through very basic form design. Each chapter in Part I may assume knowledge of and build on previous chapters. By the time you finish Part I, you can consider yourself an expert form designer and will be able to create complex form templates from scratch with no code, both for InfoPath Filler and for Microsoft SharePoint Server 2010.

If you are an application developer and already know a bit about InfoPath, feel free to skim over Part I and then jump right into Part II, where

we'll talk about advanced form design. We'll discuss adding code to your InfoPath form templates (both script and managed code), using advanced features of InfoPath Forms Services, creating add-ins, hosting InfoPath, creating custom components with ActiveX, and much more.

So, now that you know a little about InfoPath and where to go next, let's get started.



Index

Symbols

! (red exclamation point), indicating binding problem, 179

A

AAMs (alternative access mapping), 841–842

Access, cross-domain, 541–542

Access databases

choosing among ADO connection methods, 302

data types supported/unsupported for submit, 383–384

exporting data collected in mashup to, 1018–1019

InfoPath compared with, 3

Access keys

Advanced tab options, 228

in designing accessible browser-enabled forms, 800

for inserting/removing sections, 92

Access levels, user rights and, 562

Access path

alternative access mapping, 841–842

determining for form templates, 430–431

domain forms and, 536

in evaluation of best security level, 528

for forms, 449

for installable form templates, 553–555

restricted form templates and, 530–531

Accessibility tools, 800

Actions

accessing from Rules pane, 264

choosing in SharePoint workflow design, 473–474

creating, 264–267

customizing commands, 244–246

errors related to, 268–269

list of supported, 268

merge actions in aggregation namespace, 649–650

by node in merging forms, 635–636

for querying data connection, 298–299

separation between view and data and, 269–270

user roles used with, 486–488

Actions per postback setting, session state and, 825–826

Activation, of form templates, 445

Active Data Objects (ADO)

connecting to SQL Server or Access databases, 302

InfoPath using to communicate with databases, 301

Active Server Pages (ASP), 410

Active Template Library (ATL), 916

ActiveX controls

adding to Controls task pane, 950

adding to InfoPath, 904

binding options for, 953–954

in building custom controls for InfoPath, 960–962

- ActiveX controls (*continued*)
 - customizing properties, 959–960
 - data type options for, 955–958
 - enable/disable options for, 954–955
 - formatting not supported by, 206
 - in hosting clients. *See* .NET Windows Forms host application
 - inserting into form template, 959
 - installing on user's computer, 952–953
 - Internet zone and, 543
 - local intranet zone and, 539
 - overview of, 950
 - requirements for using in InfoPath, 951
 - specifying CAB file for, 951–952
 - types of custom controls, 203–204
- ActiveX objects
 - potential security risks of, 564
 - restrictions on scripting, 735
- Add Additional Details Button control, 776
- Add Custom Control Wizard
 - selecting ActiveX control for customizing, 951
 - specifying binding options, 953
 - specifying data type options, 956–958
 - specifying enable/disable property, 955
 - specifying installation options, 952
- Add-ins
 - features added for developers in InfoPath 2007, 22
 - Trust Center and, 568–570
- Add Rules button, 270–272
- Add to Favorites link, added in InfoPath 2003 SP1, 17
- Add User Roles dialog, 482–484
- Admin deployment
 - activating form template to a site collection, 809
 - activating with Manage Form Templates page, 809–811
 - activating with Site Collection Features page, 811–812
 - as alternative to user deployment, 805–806
 - command line approach to, 813–815
 - links to Forms Services, 806–807
 - managing form templates, 819–820
 - overview of, 802
 - programmatic approach to, 815–816
 - uploading form template to server, 807–808
- Administration, Forms Services
 - admin deployment. *See* Admin deployment
 - Central Administration Site, 806
 - configuring Forms Services, 821–822
 - logging, 834–841
 - managing form templates, 819–820
 - screening form templates, 816–818
 - SharePoint administrator and, 801
 - Administrative settings, data connections, 844
- Administrator-approved form templates
 - features requiring administrator approval, 803
 - location of, 808
 - publishing form template for approval, 802–805, 854–855
 - screening form templates for approval, 816–818
- Administrators
 - access levels, 562
 - privileges necessary for installing InfoPath, 526
 - publishing form template for approval by, 802–805, 854–855
- ADO (Active Data Objects)
 - connecting to SQL Server or Access databases, 302
 - InfoPath using to communicate with databases, 301
- ADO.NET datasets, 416–417
- ADOXML, 301
- Advanced controls. *See* Controls, advanced
- Advanced tab, control properties, 227–231
- Advanced tab, Form Options, 644–645
- After submit options, submitting form data and, 420–423
- Aggregate reports. *See* Reports
- Aggregation namespace, merge actions in, 649–650
- Aggregation, of controls, 954
- AJAX (Asynchronous JavaScript and XML), 107
- Alignment, text, 219, 224–225
- AllowPartiallyTrustedCallers-Attribute (APTCA), 565
- Alternative access mapping (AAMs), 841–842
- And button, specifying multiple conditions with, 252–254
- Approval workflow, in SharePoint Server 2010
 - creating, 464–466
 - description of, 463
 - using, 467–469
- APTCA (AllowPartiallyTrusted-CallersAttribute), 565

ASP (Active Server Pages), 410
ASP.NET pages, 410
Assemblies
 programmatic deployment of, 815–816
 what they are, 689
Association parameters, in SharePoint
 Server workflow, 477–478
Asynchronous JavaScript and XML (AJAX), 107
ATL (Active Template Library), 916
ATOM feeds, consuming XML data in
 InfoPath, 292
Attachments, e-mail, 517–518
Attachments field, disabling in SharePoint
 list, 1004
Attachments, file. *See* File attachment
Attributes
 adding attribute nodes, 145–146
 compared with elements, 146–147
 data source node types, 120–121
 database columns representing, 314
 default merge actions based on node type, 627
 for merge actions, 650
 merge customization by node type, 636
Authentication
 data connections and, 857
 embedded SQL authentication, 848–849
 of HTTP data connections, 847–848
 UDC authentication, 857–859
 user form templates and, 849
Authenticity, XML Signature specification, 575
Auto, Size tab options, 224
Autocomplete, Display tab options, 217
Automated workflows, 461
Automatic retrieval of data, from data
 connection, 296
Automatic security level, 527, 550–551
Automatic updates, 531
AutoText items, 656

B

Background color, example of conditional
 formatting, 261–262
Backward compatibility, Design Checker for
 checking, 104
Base64, 69
Binary format, 69
Binding data. *See* Data binding
Binding property, 961

Blank forms
 accessing form templates from, 31–32
 creating new, 34–35
Boolean (true/false) types, 125
Borders
 gridlines for, 36
 setting for Hyperlink control, 188
Breakpoints, in debugging form code, 690
Browser Compatibility messages, 793
Browser-enabled forms. *See* Forms Services
Browser optimizations
 adding Update button to augment
 postback settings, 797–799
 comments on form template performance
 in browser, 103
 controls and, 793–796
 postback settings and, 796–797
Build menu, for compiling code, 689
Bulleted List control
 binding behavior of, 115
 description and data type of, 45
 inserting, 64–66
 not supported in Forms Services, 101
Business logic, 681
Business rules, data validation based on,
 248
Button control
 Add Additional Details, 776
 binding behavior of, 115
 cannot be bound to data source, 114
 naming buttons, 752–753
 on .NET Windows Forms application,
 929–933
 overview of, 72
 Picture Button control, 73
 supported in Forms Services, 91

C

C#
 creating .NET hosting application with,
 917–918
 programming languages available to OM
 versions, 681
CAB (cabinet) files
 files contained in .xsn file, 985
 specifying for ActiveX controls, 951–952
 storing templates in, 28
 .xsn files as, 451
Cache, of form template files, 453
Calculated default values
 in fields and groups, 133–134

- Calculated default values (*continued*)
 - filtering data based on user roles, 491–494
 - vs.* static default data, 112
- Calculated Default Values section, of Rule Inspector, 274
- Calculated Value control
 - binding behavior of, 115
 - Browser Optimization messages, 794–795
 - controls supported in Forms Services, 92
 - example of use of, 191–193
 - inserting formulas, 190–191
 - overview of, 74, 189
- Cannot be blank property, fields and groups, 127–128, 132
- Cannot Be Blank rule, 249–250
- CAs (certificate authorities)
 - digitally signing form data, 575
 - getting certificates from, 555–561
- CAS (Code Access Security), 562–563
- Cascading drop-downs, List Box controls and, 358
- Caspol.exe (Code Access Security Policy Tool), 564
- Central Administration Site
 - data connection settings, 844
 - General Application Settings, 821–822
 - SharePoint settings, 806
- Centrally Managed Connection Library. *See* CMCL (Centrally Managed Connection Library)
- Certificate authorities (CAs)
 - digitally signing form data, 575
 - getting certificates from, 555–561
- Change Binding dialog, 169
- Change permissions, IRM, 609
- Change To menu item, 58–59
- Changed event
 - context-sensitive help program, 765
 - demonstrating use of, 718–723
 - overview of, 717–718
 - using in context-sensitive help program, 766–768
- Changing event
 - canceling, 705
 - limitations of, 706
 - overview of, 699
 - prompting for a password, 699–705
 - rejecting and rolling back changing node, 699
- Character formats, in Rich Text Box control, 220
- Check Box control
 - binding behavior of, 115
 - controls supported in Forms Services, 91
 - description and data type of, 44
 - not showing visuals, 179
- CheckDateTimes, for data validation, 711
- Child elements only, when submitting form data, 399–401
- Choice Group control
 - binding behavior of, 117
 - controls supported in Forms Services, 91
 - example of use of, 199–201
 - new controls in InfoPath 2003 Service Pack 1, 15
 - overview of, 197–199
- Choice nodes
 - default merge actions based on node type, 627
 - merge customization by node type, 636
- Choice Section control
 - binding behavior of, 117
 - controls supported in Forms Services, 91
 - overview of, 197–199
- Classes, not implemented by Forms Services, 788–790
- Click here to insert link, adding controls and, 51
- Close button, on Form tab of Ribbon, 89–90
- CLR (Common Language Runtime), 916
- CMCL (Centrally Managed Connection Library)
 - converting data connections to DCLs, 851–852
 - designing against, 850–851
 - domain trust form templates and, 529
 - features needing administrator approval, 803
 - getting data connection files from, 855–856
 - publishing form template for administrator approval, 854–855
 - securing cross-domain connections, 849–850
 - steps in use of, 851
 - upgrading .udcx file, 852–854
- Co-signing, digital signatures and, 598–599
- Code
 - adding business logic without writing, 247
 - adding to form templates, 686–687
 - adding to .NET hosting application, 919, 923–925
 - debugging form code, 689–690

- editing, 412
- filling out forms containing, 688–689
- form code. *See* Form code
- managed. *See* Managed code
- merging forms without writing code, 620–621
- OM (object model) and. *See* OM (object model)
- overview of, 679–680
- potential security risks of custom code, 562
- programming example. *See* Programming example (MOI Consulting request form)
- programming options in Form Options dialog, 681–685
- reference materials, 1024–1025
- upgrading, 683–684
- writing behind forms, 680
- Code Access Security (CAS), 562–563
- Code Access Security Policy Tool (Caspol.exe), 564
- Code-signing certificates, 558
- Collect Feedback workflow, in SharePoint Server 2010, 463
- Collect Signatures workflow, in SharePoint Server 2010, 463
- Collecting data. *See* Data collection
- Colors
 - new features in InfoPath 2003, 8
 - themes and, 39
- Columns
 - adding to SharePoint list, 999–1001
 - represented as attributes in databases, 314
- COM (Component Object Model)
 - add-ins category of Trust Center and, 568–570
 - creating COM object for import/export, 976
 - developers writing add-in components with, 22
 - hosting technologies for InfoPath, 916
- Combo Box control
 - binding behavior of, 114
 - description and data type of, 44
 - not supported in Forms Services, 101
 - selecting a list and, 46
- Command-line
 - admin deployment from, 813–815
 - quiescing from, 832–834
- CommandIds, enumeration of InfoPath supported commands, 933
- Commands
 - assigning shortcuts to, 244
 - customizing, 242–246
 - disabled when previewing forms, 454
 - enumeration of InfoPath supported, 933
 - updating enable/disable functionality during idle state, 935–939
- Common Language Runtime (CLR), 916
- Communication
 - host to InfoPath, 873–874, 929–933
 - InfoPath to host, 881–882
- Compatibility
 - backward compatibility, 104
 - changing Compatibility modes, 98–103
 - design modes, 96–98
 - form code with Form Services, 787–790
 - InfoPath Filler settings, 433
 - InfoPath with Web services, 328–329
 - Web browsers, 793
- Compiling code, Build menu for, 689
- Component Object Model. *See* COM (Component Object Model)
- Concatenating strings, 724
- Condition dialog, 251–255
- Conditional formatting
 - accessing from Rules pane, 258
 - adding to forms, 751
 - example specifying conditions, 259–264
 - interactive form templates with, 11–12
 - user roles and, 494–496
 - what it does, 258–259
- Conditions
 - example of use of, 253–255
 - example specifying conditional formatting, 259–264
 - making action rules conditional, 266–267
 - for pattern matching, 255–258
 - qualifiers, operands, and values when specifying, 251–252
 - specifying multiple, 252
 - user roles used with actions, 487
- Configuring
 - Forms Services, 821–822
 - services, 822
 - state service, 822–827
- Contact Selector control, 911
- Container controls
 - adding special behaviors to, 51–52
- Choice Section control, 199
- description and data type of, 45
- Horizontal Repeating Table control, 60–61

Container controls (*continued*)

- Master/Detail control, 61–64
- multiple-selection list box control, 66–67
- Optional Section control, 49–51
- Repeating Section control, 52–55
- Repeating Table control, 56–59
- Scrolling Region and Horizontal Region control, 195–197
- Section control, 45–48
- Container data types, 47
- Container nodes. *See* Groups
- Content types, SharePoint sites
 - choosing data source for, 912–913
 - custom form templates based on, 908–909
 - options in SharePoint server, 434–436
 - promoting properties with, 439
 - retrieving list of, 910–911
 - user deployment publishing as site content, 820
 - using published content type on SharePoint site, 438
- Context Changed events, 692–693
- Context-sensitive help program
 - controlling task pane focus, 764–765
 - customizing task pane for, 765–766
 - handling request connections, 772–774
 - inserting Optional Section, 777–778
 - notification of changes to views, 774–775
 - pre-query setup for `RequestType`, 769–771
 - `RequestType` event handler, 766–768
 - saving form to user's computer, 779–781
 - setting conditions on Web service queries, 768–769
 - setting up list-item approach in, 776
 - summarizing gathered data with Confirm view, 778
- ContextChanged event, controlling task pane focus, 764–765
- Control Tool Properties, Manage Rules, 248
- Controls
 - Add or Remove Custom Controls, 950
 - adding to .NET hosting application, 918–919
 - aggregation of, 954
 - binding behaviors of, 115–117
 - Browser Optimization messages, 794–796
 - changing binding behaviors, 169–170
 - changing controls to other controls, 58–59
 - conditional formatting. *See* Conditional formatting

- container controls. *See* Container controls
 - created behind data source in XSD file, 137–144
 - creating during import, 987–988
 - deleting, 117–118
 - design-time visuals on, 179
 - events, 679
 - features in InfoPath 2003, 8–9
 - features in InfoPath 2003 SP1, 14
 - form design and, 40–45
 - formatting, 205
 - Forms Services, 90–93
 - inserting, 168
 - interfaces and properties InfoPath requires, 961–962
 - List controls. *See* List controls
 - making controls fit data, 393
 - multiple binding, 178
 - new in InfoPath 2007, 21
 - new in InfoPath 2010, 24
 - Objects category of. *See* Objects
 - properties of. *See* Properties, control
 - supported in Forms Services, 91–92
- Controls, ActiveX. *See* ActiveX controls
- Controls, advanced
 - Calculated Value control, 189–193
 - choice controls, 197–201
 - hyperlinks, 186–189
 - overview of, 185–186
 - reference materials, 1022
 - Repeating Recursive Section control, 201–202
 - Scrolling Region and Horizontal Region control, 195–197
 - Vertical label control, 194–195
- Controls, custom
 - add or remove, 202–204
 - Advanced tab options, 227–231
 - changing default values, 238–242
 - customizing commands, 242–246
 - Data tab options, 210–216
 - Display tab options, 216–222
 - editing, 209–210
 - enhancements in InfoPath 2003 SP1, 14–15
 - Format Painter and, 206–207
 - formatting, 204–206
 - formatting multiple, 207–209
 - Master/Detail tab, 232–238
 - resizing multiple controls, 225–227
 - Size tab, 222–224
 - text alignment settings, 224–225

- Controls task pane
 - adding ActiveX controls to, 950
 - inserting controls with, 40–42
 - purpose and behavior of controls in, 43
 - Controls Tools Properties, 249
 - Convert Existing Form option, File tab, 967
 - Copy command
 - adding UI features to host applications, 933
 - rules, 261–262
 - Counter-signing, digital signatures and, 599–601
 - Counters, in performance monitoring, 896–899
 - Cross-domain data connections
 - administrator approval required, 803
 - CMCL (Centrally Managed Connection Library) and, 849–850
 - methods for allowing, 843
 - for user form templates, 844–846
 - Cross-domain security, data access and, 541–542
 - Custom
 - data bindings, 170–173
 - submitting form data with form code, 412–413
 - submitting form data with rules, 413–416
 - Custom (complexType) types, 125
 - Custom controls. *See* Controls, custom
 - Custom importers/exporters. *See* Import/Export framework
 - Custom merge XSL
 - creating with InfoPath, 646–647
 - overview of, 645–646
 - writing own, 647–651
 - Custom task pane
 - adding to forms, 735–736
 - controlling focus of, 766
 - in InfoPath Filler, 734
 - scripting in, 733–734, 736–739
 - simulating in browser, 785
 - Customize Commands buttons, 242–246
 - Cut command, adding UI features to host applications, 933
- D**
- Data
 - adding to SharePoint list, 1005–1006
 - allow signing entire form, 578–587
 - binding. *See* Data binding
 - defining requirements for form template, 33–34
 - editing default template, 180–183
 - form features compared with data features, 691
 - how data changes, 694
 - previewing form templates with sample data, 455–456
 - reference materials, 1021–1022
 - selecting for SharePoint workflow, 475
 - separation between view and data in
 - action rules, 269–270
 - sources. *See* Data sources
 - using digitally signed data in forms, 576–578
 - working with, 111
 - Data adapters, data connections and, 284
 - Data binding
 - for ActiveX controls, 953–954
 - advanced, 167–168
 - Binding property, 961
 - changing control binding, 169–170
 - control binding behaviors, 115–117
 - creating data source automatically, 112–114
 - customizing new bindings, 170–173
 - defined, 10, 111
 - deleting controls (but not deleting data source), 117–118
 - design-time visuals and, 178–180
 - designing forms and inserting controls, 168
 - multiple binding, 174–178
 - odd scenarios, 174
 - specifying from Binding section of Data tab, 210–214
 - Data binding, with external data sources
 - data validation, 342–343
 - designing form template, 339–341
 - filling out forms, 341–342
 - overview of, 338–339
 - Data collection
 - defining data requirements for form template, 33–34
 - InfoPath catering to, 5–6
 - Input controls and, 42
 - Data connection libraries. *See* DCLs (data connection libraries)
 - Data Connection Wizard
 - creating data connections generally, 286
 - creating database connections, 383
 - creating SharePoint connections, 288
 - data source structure in, 307
 - for databases, 303–304
 - for SQL Servers, 306
 - submit-only on main connection, 386–387

Data Connection Wizard (*continued*)
 submitting form data via e-mail, 379
 for Web services, 321, 326
 for XML files, 296–297

Data connections
 accessing form templates from data
 connection file, 32
 in browser-based forms, 334–335
 collections and, 771
 comparing Form Services with InfoPath
 client, 20
 conceptual understanding of, 283–284
 creating, 285–286
 data connection libraries (DCLs), 335–338
 deleting unused, 377
 differences between types of, 284
 handling connection failures, 772
 integrating database connections into form
 template, 312–317
 main and secondary, 287
 need for, 282
 properties and methods of, 773
 for restricted form templates, 533–534
 security prompt and, 332
 setting up database connection, 302–312
 setting up XML connection, 293–296
 SharePoint connections, 286–290
 for submitting form data, 361, 368–369
 viewing XML connection settings, 297

Data Connections dialog, 336

Data connections, Forms Services
 administrative settings, 844
 authentication to data sources, 849
 comparing Form Services with InfoPath
 client, 20
 designing against CMCL, 850–855
 e-mail connections, 862–864
 embedded SQL authentication for, 848–849
 HTTP, 847–848
 overview of, 842–843
 performance tips and best design
 practices, 893–894
 response size, 846–847
 timeouts, 846

Data-editing features, in InfoPath, 905

Data Entry Pattern dialog, 256–257

Data events

 Changed event, 717–723
 Changing event, 699–706
 choosing from list of, 686–687
 data states and, 698
 event bubbling, 694–698

 Form errors and, 714–717
 how data changes, 694
 overview of, 680, 693
 Validating event, 706–714

Data Execution Prevention (DEP), Trust
 Center, 570–571

Data integrity, XML Signature specification,
 575

Data Link files, 302

Data Source task pane (DSP), 9–10

Data sources

 adding nodes to, 145–148
 blank form starting with own data source,
 153–155
 blank form starting with XML data source,
 155–156
 blank form starting with XML Schema,
 157–158
 Cannot be blank property, 127–128, 132
 changing/reversing changes to, 119
 collections of, 771
 comparing InfoPath and Form Services, 783
 complications of starting from XML
 Schema, 158–161
 conceptual understanding of, 283–284
 converting main data source, 161–164
 creating automatically, 112–114
 data type property for fields and groups,
 124–127
 default value property, 132–134
 definitions/meanings, 120
 deleting controls (but not deleting data
 source), 117–118
 deleting nodes, 151
 Details tab, 134–136
 disabling automatic creation of, 117–118
 editing manually, 145
 enhancements in InfoPath 2003 SP1, 15
 external. *See* Secondary data sources
 field details in, 697
 finding XPath of a node, 728
 forms containing, 28
 impact on performance, 895–896
 inserting controls and, 42
 modifying properties of, 748
 moving nodes, 148–150
 name property for fields and groups,
 121–124
 namespaces in, 164–167
 operations affecting forms, 123
 organizing, 747–749
 overview of, 119–121

- pulling external data into InfoPath, 10
 - referencing nodes, 151–153
 - Repeating property, 128–131
 - secondary. *See* Secondary data sources
 - signature groups in, 592
 - traversing with XPathNavigator, 696
 - XSD file and, 137–144
- Data states
- Changed event, 717–723
 - Changing event, 699–706
 - Form errors and, 714–717
 - overview of, 698
 - Validating event, 706–714
- Data tab
- adding data connections from, 293
 - Binding section, 210–214
 - Default Value section, 214
 - Get External Data option, 285
 - Manage User Roles option, 481–482, 484
 - Receive options, 385–387
 - Submit options, 368–369, 385–387, 946
 - Validation options, 214
- Data types
- ActiveX controls and, 955–958
 - changing controls and, 58–59
 - changing in Data tab, 212–213
 - controls associated with, 42
 - fields and groups and, 124–127
 - input controls, 44–45
 - list of types available in InfoPath Designer, 125
 - nillable, 127
 - performance degrading due to large amounts of data, 309
 - SQL Server and Access, 383
 - validation errors, 126–127
 - XML Schema compared with .NET, 710
- Data validation
- accessing from Rules pane, 247–248
 - adding to forms, 751
 - advantages of InfoPath over Word, 11
 - availability only for field nodes in data source, 249–250
 - in browser-enabled forms, 126
 - Cannot Be Blank rule, 248–249
 - of forms bound to external data sources, 342–343
 - for pattern matching, 255–258
 - secondary data sources and, 770
 - specifying one or more conditions, 251–255
 - submitting form data and, 364–367
 - user roles and, 494
 - Validating event, 706–714
 - from Validation section of Data tab, 214
- Databases
- accessing form templates from, 31
 - adding tables to, 304–307
 - ADO connection methods for, 302
 - creating secondary data source using SQL queries, 309–312
 - defining relationships between tables, 308
 - InfoPath query data connection endpoints, 382
 - integrating connections into form template, 312–317
 - overview of, 301
 - previewing form based on, 315–316
 - selecting database to connect to, 304–305
 - setting up secondary data connections, 304
 - submitting form data to, 382–384
 - using connections on main and secondary data source, 303
- Date and Time Picker control
- binding behavior of, 115
 - controls supported in Forms Services, 91
 - description and data type of, 44
- Date data type, 125
- Date Picker control
- binding behavior of, 115
 - controls supported in Forms Services, 91
 - description and data type of, 44
- DateTime class, .NET, 710
- DateTime data type, 125
- DCLs (data connection libraries)
- converting data connections to, 851–852
 - cross-domain connections and, 843
 - domain trust form templates and, 529
 - overview of, 335–337
 - security and, 338
- Debugging
- Debug menu, 688–689
 - form code, 689–690
- Decimal data types, 125
- Default data
- editing, 180–183
 - static *vs.* calculated, 112
- Default value property, in fields and groups
- calculated default values, 133–134
 - static default values, 132
- Default values
- editing, 238–242
 - specifying from Default Value section of Data tab, 214

- Defense in depth, Internet zone and, 542–543
- DeleteRange method, for deleting nodes, 730–731
- DEP (Data Execution Prevention), Trust Center, 570–571
- Deploying forms. *See* Publishing
- Design a Form link, 17
- Design a Form Template dialog, 21
- Design Checker
 - backward compatibility and, 104
 - Browser Optimization messages, 793–796
 - compatibility settings for browser-enabled form template, 98–103
 - creating browser-enabled forms without using, 104
 - form code and, 781
 - maintaining compatibility between InfoPath versions, 18
 - post import warnings, 974–976, 988–990
 - reporting on browser compatibility, 793
 - switching between template compatibility modes, 96–98
- Design mode, locking files in, 453
- Design-time visuals
 - adding border to Section control, 46–47
 - data binding and, 178–180
 - features in InfoPath 2003, 10
 - field or group names in, 212
 - for master/detail controls, 234
 - showing all, 180
- Designing
 - accessible browser-enabled forms, 800
 - best practices, 889–894
 - data structure through fields task pane, 117
 - Design once approach, 95–96
 - e-mail forms, 501–504
 - form templates, 339–341
 - forms. *See* Form design, basics of
 - with InfoPath Designer. *See* InfoPath Designer
 - reports, 627–631
 - request forms, 745
 - security for form templates, 527–528
 - with SharePoint Designer 2010. *See* SharePoint Designer 2010
- Details tab
 - data connections, 297
 - data sources, 134–136
- Developer tab, 906
- Developers
 - features added in InfoPath 2010, 24–25
 - new features for in InfoPath 2007, 21–22
- Dialogs, showing a custom dialog with
 - buttons, 753–754
- Diffgrams, 301
- Digital signatures
 - co-signing with independent signatures, 598–599
 - counter-signing, 599–601
 - enabling, 100
 - enhancements in InfoPath 2003 SP1, 15
 - error checking linked to, 759–764
 - with only one signature, 594–598
 - overview of, 574–576
 - partial approach, 588–593
 - Signature Line control, 605–606
 - signed data in XML, 601–602
 - signed data in XML Schema, 602–605
 - signing entire form, 578–587
 - submitting digitally signed data, 406–409
 - user security and, 524
 - using digitally signed data in forms, 576–578
- Digitally signing form templates
 - deploying, 555–561
 - reasons for using, 561
- Disabling controls, 258
- Disabling e-mail forms, in Outlook 2010, 500
- Display tab, 216–222
- displayifs.aspx file, 995
- Disposition Approval workflow, in
 - SharePoint Server 2010, 463
- .dll extensions, 689, 815–816
- Document Information Panel
 - accessing from Developer tab, 906
 - adding controls to form templates, 912–914
 - adding form templates to, 908
 - choosing/viewing form templates, 907–908
 - creating custom form templates, 908
 - editing control properties, 915–916
 - features added in InfoPath 2007, 22
 - form templates accessed from, 32
 - inserting/editing document properties, 915–916
 - overview of, 905
 - SharePoint properties for editing Office documents, 909–912
- Document libraries
 - content types and, 908
 - editing document properties, 909

- IRM permissions, 614–618
 - for mashup pages, 1010–1011
 - merging forms from, 623–624
 - options in SharePoint server, 434–436
 - publishing to Forms Services using, 443
 - saving forms to be merged in, 621
 - submitting form data to, 374–378
 - workflows used with, 469–470
- Documents
 - choosing data source for content types in, 912–913
 - fully structured, semistructured, and unstructured, 119–120
 - inserting/editing document properties, 915–916
 - properties of. *See* Properties, document
 - SharePoint properties used to edit Office documents, 909–912
 - Word, 969
- Domain security level
 - form templates in, 537
 - Internet zone, 542–544
 - local intranet zone, 538–542
 - local machine zone, 545–546
 - moving published domain trusts, 546–548
 - overview of, 535–537
 - restricted sites, 538
 - restricting Internet zone domains, 544–545
 - security levels in InfoPath, 527
 - simulating publish domain for testing, 549
 - trusted sites zone, 545
 - which form features need, 551
- Domains
 - previewing forms in simulated domains, 458
 - what they are, 535
- Draw Table button, 13, 36
- Drop-Down List Box control
 - binding behavior of, 114
 - changing into Text Boxes, 58
 - description and data type of, 44
 - supported in Forms Services, 91
 - using with Repeating Section control, 52
- Drop-Down List Box properties dialog, 350
- DSP (Data Source task pane), 9–10
- Duplication, dealing with in list boxes, 357

E

- E-mail
 - accessing form templates from, 31
 - data connection for Forms Services, 862–864
 - data connection for restricted form templates, 533–535
 - enhancements in InfoPath 2003 SP1, 16
 - forms. *See* InfoPath e-mail forms
 - publishing forms via, 446–449
 - sending form template to e-mail recipients, 432
 - submitting form data via, 377–381
- E-Mail Attachments tab, Form Options dialog, 517–518
- E-mail forms
 - customizing e-mail support for form templates, 517–519
 - designing and using e-mail forms, 501–504
 - disabling e-mail forms in Outlook 2010, 500
 - features added to InfoPath 2007, 21
 - filling out e-mail forms, 509–511
 - forwarding, 511
 - merging and exporting, 515–517
 - opening, 513
 - overview of, 499–500
 - replying without opening, 512
 - restricted permissions required, 530
 - sorting, grouping, and filtering responses, 513–515
 - storing forms in InfoPath form folder, 500
 - storing received forms in Outlook folders, 507–509
- Edit Default Values dialog, 238–242
- Edit SQL dialog, 311–312
- editifs.aspx file, 995
- Editing
 - controls, 7, 209–210
 - data sources manually, 145
 - SharePoint lists, 998–999
- Element, data source node types, 120–121
- Embedded images, inserting, 220–221
- Embedded SQL authentication, SQL Server, 848–849
- Enable/disable options, ActiveX controls, 954–955
- Enabled property, features controls should implement, 961
- Envelope signature, 602
- Errors
 - capturing Web Service data connection errors, 774
 - deleting from Errors collection, 716
 - details of validation when working with form errors, 714–717
 - full trust template errors, 561

Errors (*continued*)

- iterating through validation errors, 708
- related to action rules, 268–269
- related to merging forms, 634
- ReportError, 710
- reporting on multiple, 713
- triggered by validation conditions, 248
- try-catch for error handling, 772
- tying error checking to digitally signing, 759–764
- in Windows Event Log, 836–838
- Errors.Add method, in event handlers, 715
- Event-based programming, in InfoPath, 693
- Event bubbling, 694–698
- Event handlers
 - Errors.Add method, 715
 - handling events on host applications, 939–941
 - for loading event, 755
 - multiple event notifications for XML event handlers, 724–727
 - .NET vs. InfoPath, 701
 - for notification events (NotifyHost method), 942–943
 - registering, 731–733
 - what they are, 692
- Event Viewer (Windows), 834–839
- EventManager object, 731–733
- Events
 - form, 679, 692–693
 - multiple event notifications for XML event handlers, 724–727
 - not implemented by Forms Services, 788–790
 - types of, 679–680
 - XML data events. *See* Data events
- Excel
 - converting Excel forms into InfoPath form templates, 963–964
 - exporting data collected in mashup to, 1018–1019
 - exporting e-mail form to, 516
 - exporting forms to, 671–675
 - hosting InfoPath client in. *See* Document Information Panel
 - Import Options dialog, 970
 - InfoPath compared with, 3–4
- Execute override method, 771, 774
- Export method, 984
- Export Source Files
 - publishing forms, 451–453
 - saving form template as source files, 646

- Export to Excel wizard, 671–675
- Export to Web, 380
- Exporters, custom. *See* Import/Export framework
- Exporting
 - collected data to Excel or Access, 1018–1019
 - forms, 670–676
 - InfoPath e-mail forms, 515–517
 - reports, 670–676
- Expression Box control. *See* Calculated Value control
- Expressions
 - functions in, 189–190
 - with values or calculations, 189
- Extensible Stylesheet Language Transformations (XSLT), 6
- External Content category, Trust Center, 571–572
- External data sources. *See* Secondary data sources
- External Item Picker control
 - binding behavior of, 115
 - controls supported in Forms Services, 92
 - description and data type of, 45

F

- Favorites link, enhancements in InfoPath 2003 SP1, 17
- Features, enabling/disabling, 364
- Field or Group properties
 - adding validation rules, 249
 - customizing merge behavior with, 631–632, 634
 - Details tab, 134–136
 - editing data sources manually, 145
 - merging forms, 631–632
- Fields
 - adding to SharePoint list, 1002–1003
 - binding controls to, 210–211
 - Cannot be blank property, 127–128, 132
 - changing names can break existing forms, 122
 - creating in DSP, 10
 - creating in SharePoint workflow design, 475–477
 - customizing merge behavior, 631–632
 - Data type property, 124–127
 - default merge actions based on node type, 627
 - Default value property, 132–134
 - as element or attribute nodes, 120–121

- inserting functions and, 193
 - leaf node placement impacting
 - performance, 174
 - merge customization by node type, 635
 - Move Field or Group dialog, 148–149
 - Name property, 121–124
 - removing from SharePoint list, 1003–1005
 - Repeating property, 128–131
 - Select a Field or Group dialog, 188, 190
 - specifying data type options for ActiveX control, 956–957
 - specifying one or more conditions, 251
 - submit options for fields and groups, 402
 - types of data source nodes, 113–114
- Fields task pane
- data source information in, 112
 - designing data structure through, 117
 - editing not allowed from, 158
- File attachment
- data type in InfoPath Designer, 125
 - overview of, 67–69
- File Attachment control
- binding behavior of, 115–116
 - file size issues, 71
 - overview of, 67–69
 - performance overhead of, 892–893
 - supported in Forms Services, 91
- File menu
- commands disabled when previewing forms, 454
 - Properties menu item, 447
 - Publish options, 427–428
 - Quick Publish option, 450
- File tab
- Convert Existing Form option, 967
 - Save & Send section, 670
- Fill Out a Form dialog, 16–17
- Filler Features tab, Form Options dialog, 676
- Filling out forms. *See also* InfoPath Filler
- binding to external data sources, 341–342
 - browser-enabled, 105–108
 - containing code, 688–689
 - determining a user's role during, 484–485
 - e-mail forms, 509–513
 - previewing compared with, 454
- Filtering
- based on user roles, 491–494
 - in InfoPath, 317
 - List Box items, 355–358
 - predicate filters, 722–723
 - responses to e-mail forms, 513–515
- Find command, adding UI features to host applications, 933–935
- Firefox, preferred browsers for use with Forms Services, 93–94
- Folders
- storing e-mail forms in, 500
 - storing received e-mail forms in, 507–509
- Footers, printing reports, 655–657
- Form code. *See also* Code
- accounting for performance in, 780
 - adding, 751–752
 - circumventing browser-enabled limitations, 783
 - compatibility with Forms Services, 787–790
 - customizing submittal of form data using, 412–413
 - data source events and, 782–783
 - debugging, 689–690
 - defined, 681
 - detecting form environment (InfoPath, Web browser, or mobile browser), 785–787
 - executing in browser, 782
 - homogeneity of programming language in, 734
 - logic as alternative to, 749
 - overview of, 781–782
 - performance tips and best design practices, 890–891
 - showing UI in browser, 784–785
 - simulating custom task pane in browser, 785
 - writing in host page, 877–881
- Form definition file (manifest), 29
- Form design, basics of
- adding controls, 40–42, 168
 - adding special behaviors to Optional Section control, 51–52
 - bulleted list, numbered list, and plain list controls, 64–66
 - Button control, 72
 - Calculated Value control, 74
 - container controls, 45
 - creating blank form template, 33–35
 - designing form layout, 35–39
 - File Attachment control, 67–69
 - Horizontal Repeating Table control added, 60–61
 - Hyperlink control, 75–76
 - Ink Picture control, 71–72
 - input controls, 42–45

Form design, basics of (*continued*)

- Master/Detail control added, 61–64
 - multiple-selection list box control, 66–67
 - Objects category of control, 67
 - Optional Section control added, 49–51
 - options for starting form design, 31–32
 - overview of, 27
 - Picture Button control, 73
 - Picture control, 70–71
 - reference materials, 1021
 - Repeating Section control added, 52–55
 - Repeating Table control, 56–59
 - Section control added, 45–47
 - Signature Line control, 76–77
 - starting InfoPath Designer, 29–31
 - testing/previewing form template, 47–48
 - themes applied for layout, 39–40
 - Vertical label control, 74–75
 - what form templates are, 28–29
- Form errors
- insufficient permissions, 432
 - types of, 715
- Form events, 679, 692–693
- Form folder, storing replies to e-mail forms, 500
- Form ID, 449, 528
- Form importers
- built-in. *See* Importers, built-in
 - custom. *See* Import/Export framework
- Form layout
- designing, 35–39
 - themes applied for, 39–40
- “Form-only” model, 363–364
- Form Options dialog
- Advanced tab, 644–645
 - automatic security level setting, 550–551
 - compatibility settings, 98
 - Digital Signatures category, 576
 - E-Mail Attachments tab, 517–518
 - Filler Features tab, 676
 - Full Trust option, 552
 - Preview tab, 490
 - Programming category, 681–685
 - Security and Trust settings, 536
- Form tab, of Ribbon
- Close button on, 89–90
 - options on, 86–87
 - Print preview button on, 90
 - Save/Save As buttons on, 89
 - Submit button on, 88
 - Update button on, 90
 - View drop-down button on, 90

Form templates

- accessing, 30–31
 - adding code to, 686–687
 - adding multiple views to, 276–278
 - browser-enabling, 98–103
 - creating blank, 33–35
 - creating custom, 908
 - customizing e-mail support for, 517–519
 - deploying digitally signed, 555–561
 - deploying installable, 553–555
 - deploying restricted, 531–535
 - deployment enhancements in InfoPath 2003 SP1, 16
 - design modes and, 96
 - designing, 339–341
 - designing browser-enabled, 95
 - exporting, 676
 - inserting ActiveX controls, 959
 - managing, 819–820
 - overview of, 4
 - previewing. *See* Previewing form templates
 - publishing browser-enabled templates, 442–445
 - publishing via e-mail, 446–449
 - quiescing, 828–830
 - restricting full trust, 562–565
 - saving and publishing, 426–429
 - screening, 816–818
 - security boundaries and, 530
 - security for, 527–528
 - structure of (manifest.xsf file), 986
 - template parts feature added to InfoPath 2007, 21
 - testing/previewing, 47–48
 - upgrading, 444
 - user roles when designing, 480
 - what they are, 28–29
- Form Warning dialog, for Form Services, 269, 418
- Format Painter
- enhancements in InfoPath 2003 SP1, 13–14
 - formatting multiple controls, 207–209
 - formatting text and controls with, 206–207
- Formatting
- controls and text, 204–206
 - controls based on conditions. *See* Conditional formatting
 - copying formats between templates, 13–14
 - multiple controls, 207–209

- FormControl class
 - methods, 920–921
 - properties, 921
- Forms
 - accessing form templates from existing, 32
 - adding custom task pane to, 735
 - data features compared with form features, 691
 - digital signing of entire form, 578–587
 - digital signing part of the form, 588–593
 - fixing imported, 972–974
 - importing into InfoPath, 967–972
 - IRM permissions on, 608–614
 - multiple digital signatures on, 586
 - performance degrading due to large amounts of data, 309
 - publishing to SharePoint, 997
 - relationship with external data source, 283
 - using digitally signed data, 576–578
 - validation occurring during loading, 713
- Forms on Office Online link, 17
- Forms Services
 - Browser forms tab, 232
 - classes, events, properties, and methods
 - not implement by, 788–790
 - Close button, 89–90
 - configuring, 821–822
 - controls, 90–93
 - creating browser-enabled forms, 104
 - creating new form, 86
 - data connections, 334–335
 - debugging in, 690
 - Design Checker, 96–98, 104
 - Design once approach, 95–96
 - designing browser-enabled forms, 95
 - disabling ambiguous parts of XML Schema, 159
 - disabling Save/Save As when submitting forms, 363
 - enabling browser-enabled forms, 98–103
 - errors in Windows Event Log, 836–837
 - filling out browser-enabled forms, 105–108
 - form code compatibility with, 787–790
 - Form tab options on Ribbon, 86–87
 - Form Warning dialog, 269, 418
 - getting started with, 79–80
 - hosting and, 419
 - hosting on Web pages, 903
 - integration with SharePoint, 801
 - leaf node placement impacting performance, 174
 - new in InfoPath 2007, 18–21
 - overview of, 80–82
 - Print Preview button, 90
 - publishing browser-enabled forms, 442–445
 - reasons for using browsers, 82–83
 - removing sections or repeating table rows, 55–56
 - Save/Save As buttons, 89
 - secondary data source binding, 339
 - stsadm.exe commands, 813–815
 - Submit button, 88
 - submit failure due to validation errors, 366
 - submitting form data via e-mail, 379, 382
 - System.Data.OleDb.NET Framework classes for database communication, 301
 - Update button, 90
 - validating browser-enabled forms, 126
 - View drop-down, 90
 - warnings in Windows Event Log, 838–839
 - Web browsers supported, 93–94
 - what it is, 83–85
- Forms Services, advanced
 - administration of, 801
 - administrative deployment, 805–816
 - alternative access mapping, 841–842
 - authentication, 849, 857
 - CMCL (Centrally Managed Connection Library) and, 849–850
 - configuring Forms Services, 821–822
 - configuring services, 822
 - configuring state service, 822–827
 - cross-domain access, 844–846
 - data connection response size, 846–847
 - data connection settings, 844
 - data connection timeouts, 846
 - data connections types, 842–843
 - designing against CMCL, 850–855
 - designing browser-enabled forms, 800
 - e-mail data connections, 862–864
 - embedded SQL authentication, 848–849
 - getting data connection files from CMCL, 855–856
 - HTTP data connections, 847–848
 - logging, 834–841
 - managing form templates, 819–820
 - optimizing controls and browsers, 792–796
 - overview of, 791
 - postback settings, 796–797
 - publishing forms, 801–802
 - publishing forms for administrator approval, 802–805

Forms Services, advanced (*continued*)
 quiescing, 827–834
 reference materials, 1025
 screening forms, 816–818
 UDC authentication, 857–859
 update button, 797–799
 Web service proxy, 859–862

Forms Services, features not supported

- Changing event, 696, 699
- class-wide variables, 704
- classes, events, properties, and methods, 788–790
- Context Changed, Merge, Save, and Sign events, 693
- date and time formats, 214
- digital signing of entire form, 408, 579
- Export Source Files, 453
- exporting forms, 670
- hiding unsupported custom controls, 204
- Horizontal Repeating Table control, 60
- InfoPath e-mail forms, 501
- Ink Picture control, 72
- IRM permissions, 607
- limitations on use of installable form templates (.msi files), 546
- Master/Detail control, 62, 232
- merging forms, 622, 645
- message box dialogs, 266, 415, 689
- multi-line checkbox, 218
- multiple controls, 55
- OCT settings, 545
- offline mode query support, 343
- placeholder text, 217
- previewing forms, 48, 455
- printing headers and footers, 656
- prompts, 267
- recursive controls, 151–152
- Repeating Choice Group, 197
- scripting, 734
- Signature Line control, 76, 268
- submitting form data to database, 383
- template parts, 203
- undo or redo, 721
- user roles, 480
- vertical text, 222
- Workflow Task message bar and workflow dialog, 470
- XML file connections, 292

Formulas

- inserting, 133–134, 189–192
- using for e-mail properties, 379
- verifying, 193

- Full Control permissions, IRM, 609
- Full trust security level
 - deploying digitally signed template, 555–561
 - deploying installable full trust template, 553–555
 - features needing administrator approval, 803
 - full trust template errors, 561
 - overview of, 552–553
 - restricting full trust templates, 562–565
 - security levels in InfoPath, 527
 - sinking Signing event, 763
 - trusted sites zone compared with, 545
- Fully trusted forms, previewing, 739–740
- Functions
 - including in expressions, 189–190
 - Insert Function dialog, 190–192
 - inserting, 133–134

G

- Gathering data. *See* Data collection
- General Application Settings, 821–822
- Getting Started dialog, InfoPath 2007, 21
- Gridlines, in layout design, 36
- Group Approval workflow, in SharePoint Server 2010, 463
- Groups
 - assigning user roles and, 483
 - Cannot be blank property, 127–128, 132
 - changing names can break existing forms, 122
 - as container nodes, 120–121
 - creating in DSP, 10
 - customizing merge behavior, 631–632
 - Data type property, 124–127
 - default merge actions based on node type, 627
 - Default value property, 132–134
 - inserting functions and, 193
 - merge customization by node type, 636
 - Move Field or Group dialog, 148–149
 - Name property, 121–124
 - Repeating property, 128–131
 - of responses to e-mail forms, 513–515
 - Select a Field or Group dialog, 188, 190
 - selecting for SharePoint workflow, 475
 - specifying data type options for ActiveX control, 956–957
 - specifying one or more conditions, 251
 - submit options for, 402
 - types of data source nodes, 113–114

H

- Handwriting recognition, Input Scope property, 228–231
- Headers, printing reports, 655–657
- Health monitoring, 900–901
- Height, Size tab options, 224
- Hiding controls, 258
- Horizontal Region control
 - binding behavior of, 116
 - overview of, 195–197
- Horizontal Repeating Table control
 - binding behavior of, 116
 - inserting, 60–61
- Host property, passing data to host using, 943–946
- Hosting controls, in Web browsers
 - host to InfoPath communication, 873–874
 - InfoPath to host communication, 881–882
 - NotifyHost, 885–888
 - overview of, 864–872
 - passing parameters from host to form, 875–877
 - SubmitToHost, 882–884
 - writing form code in host page, 877–881
- Hosting InfoPath clients
 - creating host application in .NET. *See* .NET Windows Forms host application
 - Document Information Panel. *See* Document Information Panel
 - host to InfoPath communication, 929–933
 - overview of, 904–905
 - reference materials, 1025–1026
 - submitting a form to host, 419–420, 946–949
 - technologies for, 916
- HTML, performance overhead of, 893
- HTTP
 - data connections, 847–848
 - how it works, 411
 - publishing form templates to network locations, 430
 - status codes, 391
 - submitting form data using, 409–411
- HTTPS, 411–412
- Hyperlink control
 - binding behavior of, 116
 - controls supported in Forms Services, 92
 - inserting, 186–189
 - overview of, 75–76, 186
- Hyperlink (URL) data type, 125

I

- IConversionManager interface, 982
- ID
 - form ID, 449, 528
 - LCID (locale ID), 977–978
- IE (Internet Explorer). *See also* Web browsers
 - exporting forms to, 671
 - preferred browsers for use with Forms Services, 93–94
 - security zones and, 536–537
- IFormTemplateConverter2 interface, 979–983
- IHostUtilities interface, 943–946
- IInitEventHandler, 939–941
- Import/Export framework
 - creating controls during import, 987–988
 - IFormTemplateConverter2 interface, 979–983
 - importing or exporting with, 984–986
 - overview of, 976
 - post import warnings, 988–990
 - Registry keys for importers or exporters, 977–979
 - structure of form templates and (manifest.xsf file), 986
- Import method, 984
- Import Options dialog
 - Excel workbooks, 970
 - Word documents, 969
- Import Wizard, 967–969, 971
- Importers, built-in
 - fixing the imported form, 972–974
 - importing a form into InfoPath, 967–972
 - overview of, 964–967
 - post import warnings, 974–976
- Importers, custom. *See* Import/Export framework
- Indexing SharePoint content, 801
- Indicators, in mashup pages, 1009
- Infinite loops, 727
- InfoPath 2003
 - advantages over Word, 11
 - color schemes, 8
 - conditional formatting in, 11–12
 - controls in, 8–9
 - Data Source task pane, 9–10
 - design-time visual, 10
 - enhancements in Service Pack 1, 12–17
 - layout tables in, 7
 - OM for writing scripts, 11
 - pulling external data into, 10

- InfoPath 2003 (*continued*)
 - reporting features, 12
 - Web services, 10
- InfoPath 2007
 - features added for developers, 21–22
 - features added to InfoPath client, 21
 - Forms Services introduced, 18–21
 - IRM (Information Rights Management) in, 21
 - new controls in, 21
 - overview of improvements in, 17–18
- InfoPath 2010
 - browser-enabled form capabilities improved, 23
 - compatibility with Web Services, 328–329
 - developer-related features, 24–25
 - initiating SharePoint workflow with, 470
 - integration with SharePoint. *See* Integration of InfoPath with SharePoint
 - management rules and new controls in, 24
 - performance-related features, 25
 - ribbon UI added to, 22–23
 - rights needed for installing, 526
 - split into InfoPath Designer and InfoPath Filler, 22
- InfoPath Designer
 - building custom controls, 203
 - compared with InfoPath Filler, 27
 - compatibility with Web services, 328
 - creating form templates with, 4
 - customizing merge behavior, 625, 631, 646–647
 - customizing SharePoint workflow, 478–479
 - designing browser-enabled forms, 83
 - designing forms or form templates, 28
 - drag and drop UI, 6
 - gridlines, 36
 - list of data types available in, 125
 - locking down, 572–574
 - Rule Inspector, 272–275
 - saving and publishing form templates, 426–429
 - security levels not applicable in, 529
 - splitting design features from filling features, 22, 572
 - starting, 29–31
- InfoPath e-mail forms. *See* E-mail forms
- InfoPath Filler
 - adding controls to hosting clients, 904
 - adding UI features to host applications, 929
 - custom task pane in, 734
 - Design once approach, 95–96
 - Filler Features tab, of Form Options dialog, 676
 - filling out forms, 4
 - Forms Services compared with, 80–81
 - InfoPath Designer compared with, 27
 - splitting design features from filling features, 22, 572
 - status report, 86–87
- InfoPath Forms Services. *See* Forms Services
- InfoPath, introduction to
 - benefits of, 6–7
 - data gathering and management, 5
 - integration with SharePoint, 5
 - Office user interface in, 4–5
 - overview of, 3
 - XML editing capability of, 4
- InfoPath security levels. *See* Security levels
- InfoPath Web Service, 320–327
- InfoPathControl interface, 962
- Information Rights Management (IRM). *See* IRM (Information Rights Management)
- Initiation parameters, in SharePoint Server workflow, 477–478
- Ink Picture control, 71–72
- Input controls
 - gathering data with, 42
 - list of, 44–45
 - purpose and behavior of, 43
- Input Scope, Advanced tab options, 228–231
- Insert Above, customizing command actions, 245
- Insert Below, customizing command actions, 245
- Insert, customizing command actions, 245
- Insert Formula dialog
 - adding built-in functions with, 190
 - adding Calculated Value with, 191–193
 - calculated default values and, 133–134
 - choosing field or group for, 190
 - filtering data based on user roles, 492–493
 - overview of, 189
- Insert Function dialog, 133–134, 190–191
- Insert Hyperlink dialog, 186–189
- Insert Label dialog, 194–195
- Insert Layout Table button, 13
- Insert Picture dialog, 221
- Installable form templates, deploying, 553–555
- Integers
 - formats, 213
 - types available in InfoPath Designer, 125

Integration features
 data integration in Forms Services, 80
 database connections into form templates,
 312–317

Integration of InfoPath with Microsoft
 Office, 21

Integration of InfoPath with SharePoint
 building mashup page. *See* Mashup pages
 building or updating sites, 993–994
 creating/editing SharePoint list. *See* Lists,
 SharePoint

 data connections and, 287
 designing InfoPath forms with SharePoint
 Designer, 470–471
 features in InfoPath 2010, 23
 Forms Services and, 82–83, 801
 introduction to, 5
 overview of, 993

IntelliSense information, 687

Interactive forms, action rules for, 264

InternalStartup method, for
 registering event handlers, 732

Internet addresses, Hyperlink control and,
 75–76

Internet Explorer. *See* IE (Internet Explorer)

Internet sites, trusted locations, 568

Internet zone
 overview of, 542–544
 restricting forms with OCT, 544–545

Intranet sites, trusted locations, 568

IObjectSafety interface, 961

IOleCommandTarget interface, 930–932

IPropertyNotifySink interface, 961

IRM (Information Rights Management)
 combining user roles with for security, 481
 new in InfoPath 2007, 21
 overview of, 607
 permissions on document libraries,
 614–618
 permissions on forms, 608–614
 purpose of, 524

IsBrowser, detecting form environment,
 786–787

IsMobile, detecting form environment,
 786–787

Iteration, through validation errors, 708

IViewObject interface, 962

J

JavaScript, 107

JScript
 not supported in Forms Services, 100–101

 scripting in custom task pane, 733, 739
 writing scripts with InfoPath OM, 11

K

Key fields, merge settings for, 634

L

Labels

 aligning control text with adjacent text
 labels, 393
 auto-generated, 147–148
 Vertical label control, 74–75, 115, 194–195

Languages

 creating key for supported languages
 during import/export, 977–978
 international language support, 14

Layout elements, 195–197

Layout tables

 features in InfoPath 2003, 7
 Section control compared with, 47

LCID (locale ID), for supported languages,
 977–978

Leaf nodes. *See* Fields

Libraries

 ATL (Active Template Library), 916
 Centrally Managed Connection Library.
 See CMCL (Centrally Managed
 Connection Library)

 data connection libraries (DCLs). *See* DCLs
 (data connection libraries)

 document libraries. *See* Document libraries

 getting external data from SharePoint,
 288–290

 publishing forms to SharePoint library,
 432–438

 submitting form data to SharePoint library,
 374–378

 workflows in SharePoint library, 464

Linked images, inserting, 220–221

Linux browsers, using with Forms Services,
 93–94

List Box controls

 binding behavior of, 114
 data source for populating, 349
 description and data type of, 45
 multiple-selection, 66–67
 supported by Forms Services, 91
 type-ahead capability in, 43

List Box controls, connecting to external

 data source
 cascading drop-downs, 358
 filtering items, 355–358

List Box controls, connecting to external data source (*continued*)
 overview of, 348
 pulling data from main data source, 353–354
 pulling data from secondary data source, 348–352

List controls
 bulleted list, numbered list, and plain list controls, 64–66
 controls supported in Forms Services, 92
 not supported in Forms Services, 101

List-item approach (Add Additional Details Button control), 776

List View Web Part, 1006–1007

Lists, SharePoint
 adding column to, 999–1001
 adding data to, 1005–1006
 adding field to, 1002–1003
 creating, 997
 creating for use in mashup pages, 1007–1008
 creating SharePoint file types in InfoPath, 995–996
 editing, 998–999
 example of, 997–998
 getting external data from, 288–290
 overview of, 994–995
 removing field from, 1003–1005
 submitting form data to, 371–374
 Web part, 1006–1007
 workflow design based on, 469, 472–473

LMZ (Local machine zone), 545–546

Loading event
 choosing from list of data events, 686–687
 event handlers for, 755
 Form Events, 692–693
 handling events on host applications, 939–941
 sinking, 877

Loading Form dialog, 105

Local intranet zone
 CAS permissions for, 563
 overview of, 538–542

Local machine zone (LMZ), 545–546

Locations, trusted locations category of Trust Center, 568

Locking down InfoPath Designer, 572–574

Logging
 overview of, 834
 tracing with Unified Logging Service, 839–841
 Windows Event Viewer for, 834–839

Logic
 adding without writing code, 247, 749–751
 business logic, 681

Login names, 755

Loops, infinite, 727

M

Mac OSs
 browsers for use with Forms Services, 93–94
 reasons for using browser-based forms, 82

Mail Options, for forwarding e-mail forms, 511

Main data sources
 connection types, 287
 copying data from secondary data source to, 292
 creating, 285
 database data connections on, 303
 submitting form data to Web services, 385–390
 technicalities of main and secondary data connections, 284–285

Manage Form Templates page
 in Central Administration Site, 809–811
 managing form templates with, 819–820

Manage Rules
 accessing rule options, 248
 Manage User Roles dialog, 481–482, 484

Managed code
 adding to new or existing forms, 686–687
 custom task pane and, 734
 developers writing add-in components with, 22
 enhancements in InfoPath 2003 SP1, 16
 features needing administrator approval, 803
 potential security risks of, 562
 scripting compared with, 736–737
 supported in InfoPath, 679

Management, of form templates, 819–820

Management rules, features added in InfoPath 2010, 24

Manifest (.xsf files). *See* .xsf files

Margins, Size tab options, 224

Mashup pages
 adding/configuring Web Parts, 1012–1018
 creating indicators in, 1009
 creating lists for, 996, 1007–1008
 creating new library document and page for, 1010–1011

- exporting collected data to Excel or Access, 1018–1019
 - overview of, 1006
 - SharePoint list Web part and, 1006–1007
 - Master/Detail control
 - binding behavior of, 116
 - creating master/detail relationships, 232–238
 - inserting, 61–64
 - Merge actions, in aggregation namespace, 649–650
 - Merge event, Form Events, 692–693
 - Merge Forms dialog, 622–623
 - Merge Settings dialog
 - customizing merge behavior by node type, 634–635
 - customizing merge behavior with InfoPath Designer, 646–647
 - removing blank groups, 638–639
 - separators for rich text fields, 640–642
 - showing merge customizations, 637
 - using prefixes to differentiate information in reports, 642–643
 - Merge tab, Field or Group properties dialog, 631–632
 - Merging forms. *See also* Custom merge XSL
 - actions by node, 635–636
 - alternative approaches to, 644–645
 - Control properties approach to, 632–633
 - customizing merge behavior, 625–626
 - customizing merge behavior in InfoPath Designer, 631
 - default merge actions based on node type, 627
 - errors related to, 634
 - Field or Group properties in, 631–632
 - InfoPath e-mail forms, 515–517
 - options for, 622–625
 - selecting multiple forms, 623
 - settings, 637–641
 - without writing code, 620–621
 - Message boxes
 - not supported in Forms Services, 266, 415, 689
 - showing UI with, 784–785
 - Methods
 - ADO connections, 302
 - cross-domain data connection, 843
 - data connection, 773
 - FormControl class, 920–921
 - not implemented by Forms Services, 788–790
 - OM security, 552
 - SOAP Web Service, 319
 - MFC (Microsoft Foundation Classes), 916
 - MHT (Mime HTML) file, 671
 - Microsoft Access. *See* Access databases
 - Microsoft Foundation Classes (MFC), 916
 - Microsoft Installer (.msi files), 553–555
 - Microsoft Instruction Language (MSIL), 689
 - Microsoft Maps Web Service, 330–332
 - Microsoft Office 2010
 - applications that host InfoPath. *See* Document Information Panel
 - document properties, 908
 - Excel. *See* Excel
 - Outlook. *See* Outlook
 - PowerPoint. *See* PowerPoint
 - SharePoint properties used to edit Office documents, 909–912
 - Word. *See* Word
 - Microsoft Script Editor, 736
 - Mime HTML (MHT) file, 671
 - Mobile browsers, 785–787
 - Mouse, resizing multiple controls with, 225
 - Move Field or Group dialog, 148–149
 - Mozilla. *See also* Web browsers, 93–94
 - MS Maps Web Service, 330–332
 - MSDN, documentation of InfoPath
 - supported commands, 933
 - .msi files (Microsoft Installer), 553–555
 - MSIL (Microsoft Instruction Language), 689
 - Multi-line checkbox, 218
 - Multiple bindings
 - details of, 177–178
 - Option Button control and, 179
 - overview of, 174–176
 - Person/Group Picker control and, 114
 - Multiple-Selection List Box control
 - binding behavior of, 115
 - description and data type of, 44
 - overview of, 66–67
 - Multiple selections, InfoPath Filler
 - compared with Forms Services, 93
 - Multiple views
 - creating, 276–278
 - Print dialog options, 660
 - My Data Sources folder, 304–305
 - myschema.xsd, files contained in .xsn file, 985
- ## N
- ### Names
- buttons and nodes, 752–753
 - data connections, 295

Names (*continued*)

- fields and groups, 121–124, 211–212
 - nodes, 118–119
 - SharePoint sites, 866
 - user names *vs.* login names, 755
 - user roles and, 482–484
- Namespaces
- aggregation namespace, 649
 - data sources and, 164–167, 313–314
 - XPath `NamespaceManager` parameter, 728–729
- NAT (Network Address Translation), 842
- .NET Windows Forms host application
- adding code to, 919, 923–925
 - adding controls to, 918–919
 - adding InfoPath form to Visual Studio Toolbox, 918–919
 - creating from New Project dialog in Visual Studio, 918
 - executing commands so form buttons work, 929–933
 - handling events from the form, 939–943
 - handling host notification events (`NotifyHost` method), 942–943
 - host to InfoPath communication, 929–933
 - methods available to form class, 920–921
 - overview of, 917
 - passing data to host using `Host` property, 943–946
 - properties available to form class, 922
 - saving data collected by form, 925–928
 - submitting a form to host, 946–950
 - updating toolbar buttons while application is idle, 935–939
- Netscape. *See also* Web browsers, 93–94
- Network Address Translation (NAT), 842
- Network locations, publishing forms to, 429–432
- New Project dialog, Visual Studio, 918
- `newifs.aspx` file, 995
- `nillable` data type, 127
- Nodes
- actions by node in merging forms, 635–636
 - adding, 145–148
 - changing control binding and, 171
 - choosing for data binding, 168
 - customizing merge behavior and, 631–632
 - default merge actions based on node type, 627
 - defined, 120
 - `deleteRange` method for, 730–731

- deleting, 151
 - finding XPath of, 728
 - moving, 148–150
 - namespaces, 165
 - naming, 118–119, 752–753
 - referencing, 151–153
 - selecting multiple, 729–730
 - types of, 113–114
- Nonrepudiation, XML Signature specification, 575
- `NotifyHost`
- event handler for .NET hosting application, 942–943
 - hosting controls in Web browsers and, 885–888
- Numbered List control
- binding behavior of, 115
 - description and data type of, 45
 - inserting, 64–66
 - not supported in Forms Services, 101

O

- Object model. *See* OM (object model)
- Objects, ActiveX. *See* ActiveX objects
- Objects category, of controls
- Button control, 72
 - Calculated Value control, 74
 - File Attachment control, 67–69
 - Hyperlink control, 75–76
 - Ink Picture control, 71–72
 - overview of, 67
 - Picture Button control, 73
 - Picture control, 70–71
 - Signature Line control, 76–77
 - Vertical label control, 74–75
- OCT (Office Customization Tools), 544–545
- ODC (Office Data Connection)
- picking table or view before using, 306
 - setting up database connections, 302
- Office Server connections, using HTTP
- submit, 410
- Offline mode query support
- cached queries and, 348
 - overview of, 343
 - secondary data connection types and, 344
 - setting up, 344–347
- OM (object model)
- accessing security level 3 properties and methods, 552
 - Changed event, 717–723
 - Changing event, 699–706

- data states, 698
 - DeleteRange method for deleting nodes, 730–731
 - event bubbling, 694–698
 - finding XPath of a node, 728
 - form errors and, 714–717
 - Form Events, 692–693
 - how data changes, 694
 - Internet zone and, 543
 - multiple event notifications for XML event handlers, 724–727
 - overview of, 691–692
 - programmatically admin deployment, 815–816
 - programmatically quiescing, 833
 - programming languages available to OM versions, 681
 - registering event handlers, 731–733
 - scripting and, 11, 733–734
 - security level of, 539–541
 - Validating event, 706–714
 - XML data events, 693
 - XPath NamespaceManager parameter, 728–729
 - XPathNavigator objects, 727
 - XPathNodeIterator object for selecting multiple nodes, 729–730
 - Opening screen, InfoPath Designer, 29–30
 - Operands, specifying conditions and, 251–252
 - Optimization, of Web browsers. *See* Browser optimizations
 - Option Button controls
 - binding behavior of, 115
 - controls supported in Forms Services, 91
 - description and data type of, 44
 - not showing visuals, 179
 - Optional Section control
 - access keys for inserting and removing, 92
 - adding special behaviors to form template, 51–52
 - binding behavior of, 115
 - controls supported in Forms Services, 91
 - Data tab options, 214–216
 - Edit Default Values dialog, 240
 - inserting for context-sensitive help program, 777–778
 - user formatted data and, 49–51
 - Or operator, joining conditions, 262–263
 - Outlook
 - disabling e-mail forms in, 500
 - e-mail forms in, 501
 - e-mail submit prerequisites are Outlook 2007 or Outlook 2010, 381
 - integration of InfoPath with, 21
 - publishing forms via e-mail, 447–448
 - storing received e-mail forms in Outlook folders, 507–509
 - using InfoPath e-mail forms with. *See also* E-mail forms
 - Owners, IRM permissions for creating forms, 609
- ## P
- Padding, Size tab options, 224
 - Page Design tab
 - adding multiple views to form templates, 276–278
 - creating forms based on pre-existing page layout, 36
 - themes on, 39–40
 - Page Layout Templates, 36
 - Paragraph breaks, text and, 218
 - Parameters
 - association and initiation parameters in SharePoint workflow, 477–478
 - including text and child elements only when submitting form data, 399–401
 - mapping to data sources, 399
 - passing from host to form, 875–877
 - submit options for fields and groups, 402
 - submitting form data, 398
 - Partial approach, to digitally signing form data, 588–593
 - Partial submit, of form data, 391–398
 - Paste command
 - adding UI features to host applications, 933
 - rules, 261–262
 - Paths, access. *See* Access path
 - Pattern matching, data validation for, 255–258
 - People Picker control, 92
 - PerfMon. *See* Performance monitoring
 - Performance
 - accounting for in form code, 780
 - digital signatures and, 602
 - improving by limiting amount of submitted data, 403
 - new features in InfoPath 2010, 25
 - Performance, for Forms Services
 - data connections, 893–894
 - deploying form templates, 889
 - form code, 890–891
 - form view state, 894–895

- Performance, for Forms Services (*continued*)
 - health monitoring, 900–901
 - heavy load applications, 892–893
 - miscellaneous tips, 895–896
 - overview of, 888–889
 - performance monitoring, 896–899
 - reducing/eliminating postbacks, 891–892
 - views, 889–890
- Performance monitoring
 - counters, 897–899
 - overview of, 896
- Permissions
 - error message for insufficient, 432
 - in Trustworthy Computing, 526–527
 - XML files, 291
- Permissions, IRM
 - on document libraries, 614–618
 - on forms, 608–614
- Persistence, of session data, 758–759
- Person/Group Picker control
 - binding behavior of, 115
 - description and data type of, 45
 - multiple binding, 114
- Picture Button control
 - binding behavior of, 115
 - controls supported in Forms Services, 92
 - form design and, 73
- Picture control
 - binding behavior of, 116
 - controls supported in Forms Services, 92
 - file size issues, 71
 - form design and, 70–71
- Picture data type, list of types available in InfoPath Designer, 125
- Pictures
 - attaching with Picture control, 70–71
 - Insert Picture dialog, 221
- Placeholder text, Display tab options, 217
- Plain List control
 - description and data type of, 45
 - inserting, 64–66
 - not supported in Forms Services, 101
- Post import warnings
 - built-in form importers and, 974–976
 - ImportErrors.xml file and, 988–990
- Postbacks
 - actions per postback setting, 825–826
 - Browser Optimization and, 796–797
 - postback per form settings, 825
 - reducing/eliminating, 891–892
- PowerPoint, as host. *See* Document Information Panel
- PowerShell, admin deployment with, 813
- Predicate filters, selecting nodes with XPath, 722–723
- Prefixes, to differentiate information in reports, 642–643
- Previewing
 - fully trusted forms, 739–740
 - user roles and associated actions, 488–490
 - XmlFormView, 871
- Previewing form templates
 - domain simulation, 458
 - in full trust mode, 553
 - overview of, 454–455
 - Preview button, 47–48
 - with sample data, 455–456
 - saving during preview, 458–459
 - with user roles, 456–458
- Print Multiple Views dialog, 660–661
- Print Preview button, on Form tab of Ribbon, 90
- Print Preview toolbar, 668–669
- Print Settings tab, of View Properties dialog
 - headers and footers, 656
 - print view options, 653–655, 666
 - printing multiple views, 661–662
- Print views
 - in browser-enabled forms, 668–670
 - multiple, 657–662
 - overview of, 652–655
 - Word, 662–668
- Printing reports
 - browser-enabled forms and, 668–670
 - headers and footers, 655–657
 - multiple views, 657–662
 - print views, 652–655
 - in Word, 662–668
- Privacy Options category, Trust Center, 572
- Processing instructions, XML, 404–405
- Programmatic approaches
 - to admin deployment, 815–816
 - to quiescing, 832–834
- Programming category, Form Options dialog
 - overview of, 681
 - Project location for Visual Basic and C# code, 685
 - Remove Code, 682–683
 - Upgrade OM, 683–684

- Programming example (MOI Consulting request form)
 - adding context-sensitive help. *See* context-sensitive help program
 - adding form code, 751–752
 - adding logic without writing code, 749–751
 - adding start over feature that reverts to default view, 756–759
 - designing request form, 745
 - filling out request form, 739–745
 - gathering requirements and designing visual layout, 746–747
 - making the form accessible for only one sitting, 755–756
 - naming buttons and nodes, 752–753
 - organizing data source, 747–749
 - overview of, 739
 - showing a custom dialog with buttons, 753–754
 - showing read-only properties, 754
 - tying error checking to digitally signing, 759–764
- Programming languages, options in Form Options dialog, 685
- Programming section, of Rule Inspector, 274
- Properties
 - data connection, 773
 - data source, 748
 - Field or Group, 631–632
 - form class, 922
 - not implemented by Forms Services, 788–790
 - SharePoint used to edit Office documents, 909–912
 - `XmlFormView`, 869
- Properties, control
 - accessing, 209–210
 - Advanced tab options, 227–231
 - changing control binding and, 171
 - customizing ActiveX controls, 959–960
 - customizing merge behavior, 632–633
 - Data tab options, 209–216
 - Display tab options, 216–222
 - editing controls with, 209–210
 - editing in Document Information Panel, 915–916
 - formatting, 204–206
 - Master/Detail tab, 232–238
 - Size tab options, 222–224
- Properties, document
 - inserting/editing, 915–916
 - types of, 908–909
- Properties menu item, File menu, 447
- Property promotion/demotion
 - adding, removing, modifying promoted properties outside Publishing Wizard, 439
 - defined, 436
 - property demotion, 441
 - property promotion, 438–441
 - publishing forms via e-mail and, 447
 - SharePoint library features, 433
 - using Rich Text as promoted property, 440
- Proxies, Web service, 859–862
- Publish options, File menu, 427–428
- Publishing
 - administrative deployment into Forms Services, 805–816
 - browser-enabled templates to Forms Services, 442–445
 - conveniences built into Publishing Wizard, 449–450
 - digitally signed templates, 555–561
 - form template for administrator approval, 802–805
 - form templates, 427–429
 - form templates to SharePoint, 913–914, 997
 - installable full trust templates, 553–555
 - moving published domain forms, 546–548
 - to network locations, 429–432
 - opening forms from published location, 432
 - options in SharePoint server for, 434–436
 - overview of, 425–426
 - performance tips and best design practices, 889
 - property promotion and, 438–441
 - reference materials, 1023
 - restricted form templates, 531–535
 - saving compared with, 426–427
 - to SharePoint library, 432–438
 - simulating published domain forms for testing, 549
 - trusted publishers category of Trust Center, 566–568
 - types of, 802
 - using published site content on SharePoint sites, 438
 - via e-mail, 446–449

Publishing Wizard

- convenience options available after publishing form template, 432
- conveniences built into, 449–450
- enabling browser rendering in, 443
- publishing form templates and, 430
- publishing forms to SharePoint library, 433–438
- publishing to Forms Services, 442–445
- republishing with, 449–450

Q

Qualifiers

- specifying conditions and, 251
- user roles as, 487

Queries

- compared with submit to, 385
- data connection endpoints, 382
- pre-query setup for `requestType`, 769–771
- setting conditions on, 768–769
- via rules instead of via code, 767

Quick Access Toolbar, 450, 999

Quick Publish option

- accessing on Quick Access Toolbar, 450
- attaching custom forms to workflows, 479
- publishing forms to SharePoint, 997, 999

Quick Rules

- creating rule for querying data connection, 299
- overview of, 269–272

QuickStyles, formatting text or controls with, 208

Quiescing

- applying to entire form, 831
- command-line and programmatic approaches to, 832–834
- form templates, 828–830
- overview of, 827–828
- upgrading and, 444, 831–832

R

Read, IRM permissions, 609

Read-only

- adding properties to example program, 754
- Display tab options, 217

Receive data option, Data tab, 385

Recently Used Forms link, enhancements in InfoPath 2003 SP1, 17

Recipients, publishing e-mail forms to list of, 503–504

Recursive nodes, merge customization by node type, 636

Recursive relationships, between groups, 151–152

Recursive Section controls

- new controls in InfoPath 2003 SP1, 15
- Repeating Recursive Section control and, 201–202

Red exclamation point (!), indicating binding problem, 179

Refresh button, compared with Update toolbar button, 87

Registering event handlers, with XML events, 731–733

Registry

- installable form templates in, 554
- registering importers or exporters, 977–979

Relationships, defining between tables, 308

Remove All, customizing command actions, 245

Remove, customizing command actions, 245

Removing code, programming options in Form Options dialog, 682–683

Repeating Choice Group control

- binding behavior of, 117
- overview of, 200–201

Repeating fields

- default merge actions based on node type, 627
- inserting into header or footer, 657
- merge customization by node type, 636

Repeating groups

- default merge actions based on node type, 627
- merge customization by node type, 636
- using with Section controls in merged form, 630

Repeating Numbered List control, 115

Repeating property, in fields and groups, 128–131

Repeating Recursive Section control

- binding behavior of, 116
- overview of, 201–202

Repeating Section control

- access keys for inserting and removing, 92
- adding XML fragments, 241–242
- binding behavior of, 116
- controls supported in Forms Services, 91
- converting into Repeating Table control, 57
- creating master/detail relationships, 232–238
- as Detail control, 62

- inserting multiple instances of controls
 - contained in, 52–55
 - removing, 56
- Repeating Table control
 - binding behavior of, 116
 - controls supported in Forms Services, 91
 - converting Repeating Section control to, 57–58
 - creating master/detail relationships, 232–238
 - inserting, 56
 - as Master control or Detail control, 62
 - sorting data in, 720–721
- Repetition, Repeating property for, 128–129
- Replace With, customizing command actions, 245–246
- ReportError
 - Errors.Add method compared with, 715–716
 - not using with arbitrary node, 716
 - overload errors, 710
 - removing reported errors, 714
- Reports
 - adding separators to, 640–642
 - alternative approaches to customizing merge behavior, 644–645
 - control properties approach to
 - customizing merge behavior, 632–633
 - creating custom merge XSL with InfoPath, 646–647
 - custom merge XSL, 645–646
 - customizing merge behavior, 625–626
 - customizing merge behavior in InfoPath Designer, 631
 - default merge actions based on node type, 627
 - design recommendations for, 627–631
 - exporting forms, 670–676
 - features in InfoPath 2003, 12
 - Field or Group properties approach to
 - customizing merge behavior, 631–632
 - headers and footers, 655–657
 - merge actions by node, 635–636
 - Merge Settings dialog, 637–641
 - merge settings errors, 634
 - merging forms without writing code, 620–621
 - multiple view printing, 657–662
 - options for combining data, 631
 - options for merging, 622–625
 - overview of, 619–620
 - prefixes used to differentiate information in, 642–643
 - print views, 652–655
 - print views in browser-enabled forms, 668–670
 - reference materials, 1024
 - Word print views, 662–668
 - writing own merge XSL, 647–651
- Repository
 - SharePoint library holding forms, 433
 - for Web services, 332–333
- Representational State Transfer Web Services. *See* REST (Representational State Transfer) Web Services
- Request form, in MOI Consulting example. *See* Programming example (MOI Consulting request form)
- Requirements gathering, in programming example, 746–747
- Resizing multiple controls, 225–227
- Resource Files dialog, adding XML data connections with, 293–294
- Resource Manager, 29
- Resources
 - adding resources to, 29
 - limitations on access by restricted form templates, 533–534
- Response size, data connection settings, 846–847
- REST (Representational State Transfer) Web Services
 - InfoPath query data connection endpoints, 382
 - overview of, 329–330
 - using MS Maps Web Service in InfoPath, 330–332
- Restricted security level
 - deploying restricted form templates, 531–535
 - overview of, 529–530
 - security levels in InfoPath, 527
- Restricted sites zone, 538
- Ribbon UI
 - Developer tab, 906
 - Form tab options, 86–87
 - new in InfoPath 2010, 22–23
- Rich Text Box control
 - binding behavior of, 114
 - character formats in, 220
 - description and data type of, 44
 - editing functionality of, 92–93
 - enabling browser-compatible settings, 102
 - supported in Forms Services, 91
 - user formatted data, 49

- Rich text (XHTML)
 - adding ActiveX control that consumes, 957
 - list of separators for fields, 641
 - list of types available in InfoPath Designer, 125
 - RMS (Rights Management Service)
 - required in order to use IRM with InfoPath, 607
 - specifying server for, 615
 - Roles
 - mapping SharePoint roles to IRM permissions, 617–618
 - user roles. *See* User roles
 - Rows
 - adding to SharePoint list, 1002–1003
 - inserting new row into form template, 38
 - splitting into cells, 37
 - RSS feeds
 - consuming XML data in InfoPath, 292
 - SharePoint library features, 433
 - Rule Inspector
 - categories of, 274
 - diagnosing rule-related problems, 272–273
 - Rules
 - accessing Rules pane, 248
 - action rules, 264
 - Add Rules button, 270–272
 - adding logic without writing code, 749–750
 - ADO.NET datasets, 416–417
 - applying validation rules, 708
 - conditional formatting, 258
 - creating for handling responses to e-mail forms, 505
 - customizing submittal of form data with, 413–416
 - data validation, 249
 - disadvantages for submittal of form data, 417–419
 - enhancements in InfoPath 2003 SP1, 15
 - features added in InfoPath 2010, 24
 - showing formatting rule, 260, 262
 - user roles used with action rules, 486–488
 - user roles used with conditional formatting, 496
 - user roles used with data validation, 494
 - Rules and Alerts dialog, 505
 - Rules Wizard, Outlook 2010, 505–507
- S**
- Safari. *See also* Web browsers, 93–94
 - sampledata.xml, files contained in .xsn file, 985
 - Save & Send section, of File tab, 671
 - Save/Save As buttons, on Form tab of Ribbon, 89
 - Save/saving
 - data collected by .NET hosting application, 925–928
 - disabling save functions when submitting form data, 363–364
 - dismantling form templates into files and saving, 451–453
 - Form Events, 692–693
 - form templates, 426–429
 - form templates before adding code, 686
 - forms during preview, 458–459
 - reference materials, 1023
 - submitting compared with, 362–363
 - SaveForm method, 928
 - SchemaValidation, types of form errors, 715
 - SCOM (Systems Center Operations Manager), 900–901
 - Screening form templates, for approval, 816–818
 - ScreenTip property, Advanced tab options, 227
 - ScreenTips
 - conditions and, 253
 - designing accessible browser-enabled forms, 800
 - Scripts
 - comparing Form Services with InfoPath client, 19
 - in custom task pane, 736–739
 - not supported in Forms Services, 100–101
 - overview of, 733–734
 - security risks of, 562, 564
 - writing with InfoPath OM, 11
 - Scrollbars, 195–197
 - Scrolling Region control
 - binding behavior of, 116
 - overview of, 195–197
 - Search Web Service dialog, 333–334
 - Secondary data sources
 - available/unavailable features, 341
 - binding to. *See* Data binding, with external data sources
 - connecting with. *See* Data connections
 - connection types, 287
 - copying data from main data source to, 292
 - creating, 285–286
 - databases as. *See* Databases
 - forms containing, 28

- List Box controls connected to. *See* List Box controls, connecting to external data source
- no validation in, 770
- offline mode query support, 343–348
- overview of, 281–282
- pulling external data into InfoPath, 10
- reference materials, 1022
- SharePoint as. *See* SharePoint
- sinking data events and, 720
- submitting form data to database not possible via, 382
- technicalities of main and secondary data connections, 284–285
- Web services as. *See* Web services
- XML files as. *See* XML
- Section controls
 - access keys for inserting and removing, 92
 - binding behavior of, 115
 - changing to/from Optional Section control, 215
 - grouping other controls with, 45–47
 - in merged form, 630
 - supported in Forms Services, 91
 - vs.* layout tables, 47
- Secure Sockets Layer (SSL), 847–848
- Secure Store Service (SSO), 858
- Security
 - data connection libraries (DCLs) and, 338
 - data connections and, 299
 - digital signatures. *See* Digital signatures
 - Information Rights Management. *See* IRM (Information Rights Management)
 - introduction to, 525–526
 - locking down InfoPath Designer, 572–574
 - overview of, 523–524
 - reference materials, 1023–1024
 - of submit using HTTPS, 411–412
 - Trust Center. *See* Trust Center
 - user roles are not a security feature, 481
- Security and Trust settings
 - in Form Options dialog, 536
 - setting up data connections and, 293
- Security levels
 - automatic, 550–551
 - deploying digitally signed template, 555–561
 - deploying installable full trust template, 553–555
 - deploying restricted form templates, 531–535
 - designing security for form templates, 527–528
 - domain level, 535–537
 - full trust security, 552–553
 - full trust template errors, 561
 - Internet zone, 542–544
 - local intranet zone, 538–542
 - local machine zone, 545–546
 - moving published domain trusts, 546–548
 - overview of, 526–527
 - restricted level, 529–530
 - restricted sites, 538
 - restricting full trust form templates, 562–565
 - restricting Internet zone domains, 544–545
 - simulating publish domain for testing, 549
 - trusted sites zone, 545
- Security zones
 - determining, 536–537
 - in domain security, 535
 - Internet, 542–544
 - local intranet, 538–542
 - restricted sites, 538
 - trusted sites, 545
- Select a Field or Group dialog
 - creating Web Service data connection, 325
 - inserting formulas and, 190
 - inserting functions and, 193
 - inserting hyperlinks and, 188
 - specifying one or more conditions, 251
- Send data to the server dialog, Browser Optimization and, 796
- Send Data to Web Part, in Forms Services, 80
- Sender, as object parameter, 697
- Sensitive data, not storing in main data source, 748
- Separators, adding to reports, 640–642
- Sequence nodes
 - default merge actions based on node type, 627
 - merge customization by node type, 636
- Sequential workflows, 462–463
- Servers
 - form features that communicate with automatically, 106–107
 - SharePoint. *See* SharePoint Server
 - SQL Server. *See* SQL Server
 - Web servers. *See* Web servers
- Services
 - configuring, 822
 - state service, 822–827
- Session data, persistence of, 758–759
- Session state
 - actions per postback, 825–826
 - configuring state service, 822–823
 - in Forms Services, 823–824

- Session state (*continued*)
 - how sessions can be destroyed, 824–825
 - maximum size setting, 826–827
 - performance and, 894–895
 - postbacks per form, 825
 - terminate active sessions, 826
- Session state service (SSS), 894–895
- Shared folders, including when publishing form templates, 429
- SharePoint Designer 2010
 - association and initiation parameters in determining when to request information in workflow, 477–478
 - choosing workflow actions, 473–474
 - Create Reusable Workflow dialog for data collection, 472–473
 - creating fields for workflow, 475–477
 - designing workflows with, 463, 470–471
 - opening interface, 472
 - publishing workflow to, 478
 - selecting data for workflow, 475
 - selecting users or groups for workflow, 474–475
- SharePoint Foundation 2010, 462–463
- SharePoint Server
 - accessing form templates from, 31
 - in administration of Forms Services, 801
 - Central Administration Site, 806
 - connections, 286–290
 - customizing forms with Forms Services, 82
 - designing forms with InfoPath, 22
 - document libraries. *See* Document libraries
 - document properties, 908–909
 - file attachments in, 67
 - form templates based on content types, 908
 - Forms Services built on Server version 2007, 20
 - InfoPath query data connection endpoints, 382
 - integration with Forms Services, 82–83, 801
 - integration with InfoPath. *See* Integration of InfoPath with SharePoint
 - lists. *See* Lists, SharePoint
 - publishing form template to, 913–914
 - publishing forms to SharePoint library, 432–438
 - submitting form data to, 374–378
- SharePoint Server workflows
 - association and initiation parameters, 477–478
 - checking on status of workflow requests, 470
 - choosing SharePoint site for workflow design, 471
 - choosing workflow actions, 473–474
 - creating fields for, 475–477
 - creating workflows based on lists, 472–473
 - customizing with InfoPath Designer, 478–479
 - default set, 463–464
 - designing with SharePoint Designer or Visual Studio, 470
 - example creating Approval workflow, 464–466
 - example using Approval workflow, 467–469
 - initiating from within InfoPath, 470
 - selecting data for, 475
 - selecting users or groups for, 474–475
 - XML form associated with workflow, 477
- SharePoint sites
 - activating form template to site collection, 809
 - building or updating, 993–994
 - choosing in workflow design, 471
 - content types. *See* Content types, SharePoint sites
 - indexing content on, 801
 - naming, 866
 - securing, 820
 - site collections, 442
 - using published site content on, 438
- Shortcuts, assigning to commands, 244
- Sign event, Form Events, 692–693
- Signature groups, in data sources, 592
- Signature Line control, 76–77, 605–606
- Signing event
 - linking digital signature to errors, 762–763
 - sinking, 762
- Sinking data events
 - choosing from list of data events, 686–687
 - custom task pane and, 735
 - Loading event, 877
 - secondary data sources and, 720
 - Signing event, 762
 - Validating event, 708, 712
- Site Collection Features page, Central Administration Site, 811–812
- Site collections, SharePoint, 442
- Site Content Type. *See* Content types, SharePoint

- Sites, SharePoint. *See* SharePoint sites
- Size tab
 - customizing controls, 222–224
 - making controls fit data, 393
- SOAP Web Service
 - connecting via URL, 317–318
 - creating in Visual Studio 2010, 318
 - description of methods, 319
 - InfoPath query data connection endpoints, 382
 - navigating to, 319
 - searching with UDDI, 333–334
 - test page, 320
 - using InfoPath as client, 320–327
 - Web service compatibility with InfoPath, 328–329
- Sorting
 - data in repeating Table, 720–722
 - in InfoPath, 317
 - query sort order, 309–310
 - responses to e-mail forms, 513–515
 - view-based, 719
- Source code, reviewing before deployment, 818
- Source forms, in merging forms, 625
- SP File Attachment, 92
- Speech recognition, Input Scope property, 228–231
- Spellchecking, Display tab options, 217
- Split Table dialog, 37
- Spreadsheets, exporting promoted data to, 433
- SQL Server
 - choosing among ADO connection methods, 302
 - data types supported and unsupported for submit, 383–384
 - embedded SQL authentication, 848–849
 - InfoPath native support for SQL Server 2002 and later, 301
 - session state bottlenecks, 894–895
- SQL statements
 - applying SQL queries to create secondary data source, 309–312
 - query sort order, 309
 - testing, 311–312
- SSL (Secure Sockets Layer), 847–848
- SSO (Secure Store Service), 858
- SSS (session state service), 894–895
- Start over feature, adding to example program, 756–759
- State, data. *See* Data states
- State machine workflows, 462–463
- State service, configuring, 822–827
- Static default values
 - in fields and groups, 132–134
 - vs.* calculated default data, 112
- Storing e-mail forms
 - in InfoPath form folder, 500
 - received forms in Outlook folders, 507–509
- Strings
 - concatenating, 724
 - list of types available in InfoPath Designer, 125
- stsadm.exe
 - admin deployment with, 813–815
 - quiescing commands with, 833
- Submit button, on Form tab of Ribbon, 88
- Submit data connections
 - overview of, 361
 - types of, 368–369
- Submit data option, of Data Connection Wizard, 385–387
- Submit event, Form Events, 692–693
- Submit Options dialog
 - Advanced section of, 421
 - HTTP submit, 409
 - setting up submit for form template, 368–369
 - submitting form to host, 946
- Submitting forms
 - ADO.NET datasets and rules, 416–417
 - After submit options, 420–423
 - customizing using form code, 412–413
 - customizing using rules, 413–416
 - data validation and, 364–367
 - to databases, 382–384
 - digitally signed data and, 406–409
 - disabling Save/Save As and, 363–364
 - disadvantages of rules for, 417–419
 - entire form, 402–405
 - to host application, 946–950
 - to hosting environment, 419–420
 - including text and child elements only, 399–401
 - overview of, 361
 - parameters, 398
 - partial submit, 391–398
 - process of, 367–368
 - reasons for, 362–363
 - reference materials, 1022
 - saving form to user's computer, 779–781

Submitting forms (*continued*)
 to SharePoint library, 374–378
 SharePoint lists, 371–374
 types of methods for, 369–370
 types of submit data connections, 368–369
 via e-mail, 377–381
 to Web Server via HTTP, 409–411
 to Web Server via HTTPS, 411–412
 to Web Services, 384–390
 XML subtrees in partial submit, 401–402
 SubmitToHost, 882–884
 SystemGenerated errors, 715
 Systems Center Operations Manager
 (SCOM), 900–901

T

Tab index
 Advanced tab options, 227–228
 designing accessible browser-enabled
 forms, 800
 Table Tool Layout tab, 35–36
 Tables
 adding to databases, 304–307
 allowing multiple records to be displayed
 in a form, 310
 defining relationships between, 308
 inserting, 220–221
 layout tables introduced in InfoPath 2003, 7
 modifying behavior of, 309
 selecting entire, 38
 tools for making table creation easier, 13
 Tables toolbar, enhancements in InfoPath
 2003 SP1, 13
 Tablet PCs
 enhancements in InfoPath 2003 SP1, 16
 Ink Picture control and, 71–72
 Target forms, in merging forms, 625
 Task panes
 controls. *See* Controls task pane
 custom. *See* Custom task pane
 Data Source, 9–10
 fields. *See* Fields task pane
 hiding *vs.* disabling, 775
 XML Structure task pane, 663–664
 TCS (Trusted Certificate Store), 555–561
 Template parts
 accessing form templates from, 32
 adding ActiveX controls to Controls task
 pane, 950
 types of custom controls, 203
 template.xml, files contained in .xsn file, 985

TerraServer map services, 330–332
 Text
 aligning control text with adjacent text
 labels, 393
 alignment, 219, 224–225
 containers for, 49
 entering multiple lines of, 218
 formatting, 204–205
 formatting using Format Painter, 206–207
 including Text elements when submitting
 form data, 399–401
 QuickStyles, 208
 vertical display of. *See* Vertical label
 control
 Text Box control
 binding behavior of, 114
 description and data type of, 44
 limiting number characters in, 219
 Size tab, 223–224
 supported in Forms Services, 91
 using for address data, 46
 Text selection
 formatting multiple controls, 207
 formatting text and, 205
 Themes, for form layout, 39–40
 Time data type, list of types available in
 InfoPath Designer, 125
 Time-sensitive/critical request types, 753
 Timeouts, data connections, 846
 Toolbars
 hiding in Web pages hosting forms, 87
 updating toolbar buttons during idle state,
 935–939
 ToolStrip control, Visual Studio 2005, 929
 Trace logs, using Unified Logging Service,
 839–841
 Translation Management workflow, in
 SharePoint Server 2010, 463
 Trust, as component of security, 523
 Trust Center
 add-ins category, 568–570
 DEP setting category, 570–571
 External Content category, 571–572
 list of trusted publishers, 556
 overview of, 566
 Privacy Options category, 572
 trusted locations category, 568
 trusted publishers category, 566–568
 Trust settings, setting up data connections
 and, 293
 Trusted Certificate Store (TCS), 555–561

Trusted locations category, Trust Center, 568
Trusted publishers category, Trust Center, 566–568
Trusted sites zone, 545
try-catch, in error handling, 772
TWC (Trustworthy Computing)
 overview of, 525
 requests for permissions and granting permissions, 526–527
Type-ahead capability, in List Box controls, 43

U

UDC authentication, 857–859
.udcx file, 852–854
UDDI, searching Web Services with, 333–334
UI (user interface)
 comparing Form Services with InfoPath client, 19
 for controls, 40
 Ribbon UI added to InfoPath 2010, 22–23
 showing UI with message boxes, 784–785
ULS (Unified Logging Service), 839–841
Uniform Resource Locations (URLs). *See* URLs (Uniform Resource Locations)
Uniform Resource Name (URN), 528
UNIX OSs
 browsers for use with Forms Services, 93–94
 reasons for using browser-based forms, 82
Update button
 augmenting postback settings, 797–799
 on Form tab of Ribbon, 90
 Refresh button in browser compared with, 87
UpDown Control, example of use of
 ActiveX control in InfoPath, 960
Upgrading code, programming options in
 Form Options dialog, 683–684
Upgrading form templates, quiescing and, 444, 831–832
Upload Form Template page
 in Central Administration Site, 807–808
 quiescing and, 831–832
URLs (Uniform Resource Locations)
 AAMs (alternative access mapping), 841–842
 access path for form template, 528
 connecting to Web Services with, 317–318
 Hyperlink control and, 75–76
 Hyperlink data type, 125
 identifying domain forms by, 535

URN (Uniform Resource Name), 528
Usability, designing browser-enabled forms, 800
User accounts, access levels for, 562
User deployment
 admin deployment as alternative to, 805–806
 publishing as site content, 820
User interface. *See* UI (user interface)
User login, 861
User names
 assigning user roles and, 482–484
 vs. login names, 755
User publishing, 802
User roles
 action rules and, 486–488
 conditional formatting with, 494–496
 creating, 481–484
 data validation with, 494
 determining a user's role when filling out forms, 484–485
 filtering data based on, 491–494
 InfoPath e-mail forms and, 502
 not available in browser-enabled forms, 84
 overview of, 479–481
 previewing, 488–490
 previewing form templates with, 456–458
 private and public, 15
 views based on, 497–499
UserDefined errors, 715
Users, selecting for SharePoint workflow, 475

V

Validating event, 706–714
Validation
 Data tab options, 214
 data types, 126–127
 Rule Inspector and, 274
 rules. *See* Data validation
 signatures dialog and, 742
Values, specifying conditions and, 252
VBScript, 11, 733, 739
VeriSign, trusted certificates from, 559
Version Upgrade events, Form Events, 692–693
Vertical label control
 binding behavior of, 115
 inserting, 194–195
 not supported in Forms Services, 102
 overview of, 74–75
View drop-down, on Form tab of Ribbon, 90

View Properties dialog
 adding multiple views to form templates, 277–278
 Print Settings tab, 653–656
 Text Settings tab, 208–209

View state, 894–895

View *switched* events, Form Events, 692–693

view1.xsl, files contained in .xsn file, 985

Views
 confirming display summary of gathered data, 778
 creating multiple, 276–278
 form templates, 28–29
 notification of changes to, 774–775
 performance tips and best design practices, 889–890
 print views. *See* Print views
 role-based, 497–499
 separation between view and data in action rules, 269–270
 setting default, 757
 switching between, 278

Virtual controls, Choice Group control, 198

Visio Services, adding diagrams to workflows, 470

Visual Basic
 InfoPath compared with, 4
 programming languages available to OM versions, 681

Visual Studio
 adding InfoPath form to Visual Studio Toolbox, 918–919
 creating Web Service with, 318
 designing workflows with, 470
 New Project dialog, 918
 ToolStrip control in Visual Studio 2005, 929

VSTA (Visual Studio Tools for Applications) development environment, 687
 features added for developers in InfoPath 2007, 22
 previewing forms, 688–689
 writing form code with, 679

VSTO (Visual Studio Tools for Office), 21

W

Waivers, data validation, 711–712

Warnings, in Windows Event Log, 838–839

Web browsers. *See also* Forms Services
 browser-enabled form capabilities improved in InfoPath 2010, 23
 browser optimizations, 103

circumventing browser-enabled limitations, 783

compatibility issues and, 84–85, 793

controls not available when designing browser-enabled forms, 40–41

detecting as form environment, 785–787

disabling browser rendering, 803

executing form code in, 782

Forms Services supported, 93–94

hosting controls in. *See* Hosting controls, in Web browsers

InfoPath forms in, 80–82

mobile browsers, 785–787

not able to preview browser-enabled forms in, 48

optimizing. *See* Browser optimizations

reasons for using browser-based forms, 82–83

showing UI with faux message boxes, 784–785

simulating custom task pane in browser, 785

Web designers, features of Forms Services benefiting, 20

Web Front End (WFE), 865

Web Parts
 adding/configuring for mashup pages, 1012–1018
 adding/configuring for Wiki Page library, 1012–1018

SharePoint 2007 and, 994
 for SharePoint lists, 1006–1007

Web servers
 publishing form templates to, 429–430
 submitting form data to, 409–412

Web Service Definition Language. *See* WSDL (Web Service Definition Language)

Web service proxy, 859–862

Web services
 accessing form templates from, 31
 compatibility with InfoPath, 328–329
 features in InfoPath 2003, 10
 handling request connections, 772–774
 repository for, 332–333
 REST Web Services, 329–332
 searching with UDDI, 333–334
 setting conditions on event querying, 768–769
 submitting form data to, 384–390
 for user deployment, 820
 using InfoPath Web Service, 320–327
 using SOAP Web Service, 318–320

- what they are, 317–318
- WFE (Web Front End), 865
- Width, Size tab options, 224
- Wiki Page library
 - adding/configuring Web Parts for, 1011–1012
 - creating, 1010–1011
- Windows Forms applications. *See* .NET Windows Forms host application
- Windows OSs
 - browsers for use with Forms Services, 93–94
 - Event Viewer (Windows), 834–839
- Word
 - converting Word forms into InfoPath form templates, 963–964
 - exporting forms to, 671
 - fixing imported form, 972–974
 - hosting InfoPath client in. *See* Document Information Panel
 - Import Options dialog, 969
 - importing form from, 965–971
 - InfoPath 2003 advantages *vs.*, 11
 - InfoPath compared with, 3–5
 - post import warnings, 974–976
 - print views in, 662–668
- WordPrint tool, 663
- Workflow dialog, 469
- Workflow, e-mail forms
 - customizing e-mail support for form templates, 517–519
 - designing and using e-mail forms, 501–504
 - disabling e-mail forms in Outlook 2010, 500
 - filling out e-mail forms, 509–513
 - forwarding, 511
 - merging and exporting, 515–517
 - opening, 513
 - overview of, 499–500
 - replying without opening, 512
 - sorting, grouping, and filtering responses, 513–515
 - storing forms in InfoPath form folder, 500
 - storing received forms in Outlook folders, 507–509
- Workflow, generally
 - automatic *vs.* manual initiation of, 464–465
 - creating using graphical designer, 463
 - enhancements in InfoPath 2003 SP1, 16
 - for merging forms into report, 621
 - overview of, 461–462
 - reference materials, 1023

- sequential and state machine types, 462–463
- SharePoint library features and, 433
- what they are, 462

- Workflow Settings, SharePoint library, 464
- Workflow, SharePoint Server 2010
 - association and initiation parameters, 477–478
 - automatic *vs.* manual initiation of, 464–465
 - checking on status of workflow requests, 470
 - choosing SharePoint site for workflow design, 471
 - choosing workflow actions, 473–474
 - creating fields for, 475–477
 - creating workflows based on lists, 472–473
 - customizing with InfoPath Designer, 478–479
 - default set, 463–464
 - designing with SharePoint Designer or Visual Studio, 470
 - example creating Approval workflow, 464–466
 - example using Approval workflow, 467–469
 - initiating from within InfoPath, 470
 - selecting data for, 475
 - selecting users or groups for, 475
 - XML form associated with workflow, 477
- Workflow, user roles and, 479–499
 - action rules and, 486–488
 - conditional formatting with, 494–496
 - creating, 481–484
 - data validation with, 494
 - determining a user's role when filling out forms, 484–485
 - filtering data based on, 491–494
 - overview of, 479–481
 - previewing, 488–490
 - views based on, 497–499
- Wrapping text, 218
- WSDL (Web Service Definition Language)
 - connecting to Web Services with WSDL URL, 318
 - overview of, 317
 - portability of, 322

X

- X509, digital signature standard, 602
- XDocument.Roles property, in
 - determination of user roles, 486

- XHTML (Rich text)
 - adding ActiveX control that consumes, 957
 - list of separators for fields, 641
 - list of types available in InfoPath Designer, 125
- XMethods Web site, 332–333
- XML
 - accessing form templates from, 31
 - advantages of InfoPath over Word, 11
 - benefits of, 6
 - blank form starting with XML data as data source, 155–156
 - converting main data source, 161–164
 - data connection for restricted form templates, 533–534
 - data events. *See* Data events
 - digitally signing XML data, 601–602
 - file data adapter, 290–292
 - format for attached files, 69
 - fragments, 239–241
 - as fully structured document, 119–120
 - inferring, 157
 - InfoPath query data connection endpoints, 382
 - multiple event notifications for XML event handlers, 724–727
 - nodes in, 120
 - processing instructions, 404–405
 - setting up data connection, 293–296
 - submitting digitally signed data, 406–409
 - using XML file data on form, 297–300
 - viewing data connection settings, 297
 - XML file adapter, 290–292
 - XML form associated with SharePoint workflow, 477
 - .xsn files and, 426
- XML Schema
 - accessing form templates from, 31
 - addition of ambiguous schemas not permitted, 164
 - converting main data source, 161–164
 - converting to ambiguous schemas not permitted, 163
 - created behind data source in XSD file, 137–144
 - as data source, 157–158
 - digitally signing, 602–605
 - DSP and, 9–10
 - for example form template, 135–136
 - inferring, 157
 - potential complications of starting from, 158–161
 - rules binding XML structure, 9
 - structured and unstructured documents and, 119–120
 - .xsn files and, 426
- XML Signature
 - digitally signing with, 574
 - W3C specification for, 575
- XML Structure task pane, 663–664
- XML subtrees, in partial submit, 401–402
- XmlFormView control
 - initializing, 874
 - overview of, 869
 - previewing, 871
 - properties of, 869, 879
 - writing code into host pages, 877–878
- XPath (XML Path)
 - finding node's, 728
 - NamespaceManager parameter, 728–729
 - predicate filters, 722–723
 - processing XML data, 6
 - registering event handlers, 733
 - viewing for formula field, 190
- XPathNavigator objects
 - AppendChildElement method, 778
 - override behavior, 771
 - using, 727
 - using with Validating event, 709
 - what they are, 696
- XPathNodeIterator object, for selecting multiple nodes, 729–730
- .xsd files (XML Schema)
 - controls and XML Schema created behind data source in XSD file, 137–144
 - defining XML rules, 6
 - extracting from .xsn file, 451–452
- .xsf files
 - extracting from .xsn file, 452–453
 - limitations on resource access by restricted form templates, 533
 - persistence of publishing options in, 450
 - .xsn files and, 426
- .xsf (manifest) file, 426
 - extracting from .xsn file, 452–453
 - files contained in .xsn file, 985
 - as form definition file, 29
 - limitations on resource access by restricted form templates, 533
 - making changes to, 646–647
 - persistence of publishing options in, 450
- xsi:nil, 724
- .xsl files (view)
 - custom merging. *See* Custom merge XSL
 - extracting from .xsn file, 451–452



- view-based sorting, 720–722
- Word print views and, 665–666
- .xsn files and, 426
- XSLT (Extensible Stylesheet Language Transformations), 6
- .xsn files
 - breaking down into component files, 451–453
 - digitally signing, 555

- e-mail forms and, 518
- file extension for form templates, 28
- files contained in, 985
- publishing form templates to network
 - locations, 429
- saving and, 426

Z

Zones. *See* Security zones